ELFIN Mission Operations and ADCS Design

Ethan Tsai *, Sophie Ye †, Austin Norris ‡, Cindy Russell §, Jiashu Wu ¶, and Vassilis Angelopoulos ‖
Experimental Space Physics Lab at the UCLA Department of Earth, Planetary, and Space Sciences, Los Angeles, CA 90095

Akhil Palla **
Reliable Robotics, Mountain View, CA 94043

Jason Mao ††
DEPT, Newburyport, MA 01950

Sharvani Jha ‡‡ and Chanel Young §§
Microsoft, Redmond, WA 98052

James King ¶¶
Plaid, San Francisco, CA 94103

The Electron Loss and Fields INvestigation (ELFIN) mission comprises of two 3U+ space weather CubeSats developed, built, and operated by several generations of undergraduate students at UCLA. These CubeSats uniquely spin at 21 RPM and are designed to produce novel measurements of precipitating, trapped, and reflected populations of electrons and ions in the radiation belts. ELFIN was on orbit for four successful years before deorbiting in September 2022. It is now recognized as one of the most scientifically productive CubeSats in the NASA and NSF CubeSat fleet with over 40 refereed scientific publications in less than 3 years. However, ELFIN mission operations had a challenging start and only became successful after a complete refactor of the mission's operational paradigm in early-2020. The refactor yielded significantly higher data downlink volume, minimized human error, and required far fewer hours on console. This was achieved not by automating satellite operations, but rather by defining a set of abstractions with rigid rules. We present this new operational framework (including implementation of our onboard flight software and Attitude Determination and Control System (ADCS) design) that allowed a small team of students to achieve reliable daily

*Project Scientist
†Undergraduate Student Researcher
‡Software Engineer
§Programmer
¶Researcher
‖Professor
**Flight Software Engineer
††Software Engineer
‡‡Software Engineer
§§Software Engineer
¶¶Software Engineer

operations for two satellites. Strategies implemented here are designed with the idea of scaling to multi-satellite operations of science observatory missions. Finally, we will highlight the scientific progress enabled by these new ELFIN data sets, which emphasize the role of electron precipitation measurements for future studies in magnetospheric, ionospheric, and atmospheric physics.

Nomenclature

(Nomenclature entries should have the units identified)

α = pitch angle, angle between particle velocity and local magnetic field

R_e = Earth radius, 6371 km

L-shell = refers to the set of magnetic field lines which cross the Earth's magnetic equator at LR_e

$\Delta E/E$ = ratio of energy bin size to the center of the energy bin, which describes the energy resolution

I. Introduction

Earth's magnetic field traps energetic particles into massive toroidal regions around Earth called the radiation belts. Electrons traveling at relativistic velocities (> 500 keV) are so dynamic that they can vary by five orders of magnitude on timescales that span from days to minutes [1]. Such electrons can pose a risk to spacecraft and astronaut safety and are studied under the umbrella term of "space weather" [2, 3]. Energetic electrons (> 100 keV) can leave the radiation either diffusely (back out into the magnetosphere) or via precipitation, where electrons are field-aligned enough (at given energies) such that they collide with the denser atmosphere and become lost (typically around ~ 100 km) at high latitudes, sometimes correlated with auroral phenomena [4, 5]. There remain many open questions regarding the nature of electron dynamics of electron energization or losses, both of which are currently still impossible to predict during some of the most active magnetospheric phenomena: geomagnetic substorms and storms. It is important to study these aspects, as they are crucial to the field of space weather, which focuses on forecasting risks to protect society's reliance on space technologies. While there have been many spacecraft missions dedicated to exploring the equatorial regions of the magnetosphere (because many magnetospheric processes take place here), there has yet to be a mission dedicated to providing high-quality measurements of electron losses at high cadences with global coverage which necessarily requires a low-altitude, polar orbitting spacecraft.

The Electron Loss and Fields INvestigation with a Spatio-Temporal Ambiguity-Resolving (officially ELFIN-STAR, but also ELFIN* or, commonly, ELFIN) mission was designed to fill this niche [6]. The mission consists of two 3U+ CubeSats that were developed, built, and operated by several generations of undergraduate students at UCLA. Both ELFIN CubeSats launched together on September 15th, 2018 into a ~450 km circular polar orbit with a 93° inclination.

After 4 fulfilling years on orbit, ELFIN-A re-entered on September 17th, 2022 and ELFIN-B re-entered on September 30th, 2022, thereby concluding the operational phase of the mission. ELFIN was designed to capture, for the first time, the energy distributions of energetic electrons and ions with pitch angle (α, the angle between particle velocity and magnetic field) resolution high enough to resolve the bounce loss cone (particles with pitch angles within the loss cone are lost within one bounce period). The primary objective was to determine the storm-time particle precipitation rates and elucidate the variety of different mechanisms that can lead to such particle losses. ELFIN accomplished this using an Energetic Particle Detector for Electrons and Ions (EPDE, EPDI) – capable of measuring the energy and pitch-angle distributions of energetic electrons and ions with $\Delta E/E = 40\%$ across 16 energy channels between 50 keV and 5 MeV – along with a Fluxgate Magnetometer (FGM) deployed at a distance of ∼75 cm to avoid noise from the spacecraft bus. By spinning at just over 21 revolutions per minute (spin period $\approx$ 2.8 sec) with its spin plane aligned with the orbital plane, ELFIN's 16 sectors per spin yielded a spin phase resolution of $\Delta\alpha = 22.5^\circ$ and allowed for full pitch angle coverage with sub-loss cone resolution. From its low-altitude vantage point, ELFIN turned ON its instruments to collect data as it traversed the L-shell ranges $3 < L < 18$, providing a radial snapshot of equatorial processes at a given magnetic local time (MLT) in what we called a "Radiation Belt Crossing" or "Science Zone" (SZ).

This was unique because no previous mission had been able to achieve such measurements of precipitating particle distributions within the outer radiation belt. The loss cone during these science zones collections is around $\sim 65^\circ - 70^\circ$ – whereas it usually is $< 3^\circ$ at the equator – so equatorial missions, like the Time History of Events and Macroscale Interactions during Substorms (THEMIS) [7], Magnetospheric Multiscale (MMS) [8], and Van Allen Probes (VAP) [9] cannot easily resolve any spectral details inside it. Although equatorial, the Exploration of energization and Radiation in Geospace (ERG) [10] mission has the ability to measure precipitating electrons from the equator with its high pitch angle resolution. However, these electron precipitation measurements are limited in energy to < 100 keV due to its extremely narrow field of view (FOV) [10]. Unfortunately, this does not allow for the study of how particles can be accelerated to such relativistic energies and how off-equatorial processes may affect precipitation rates. There have been a number of low-altitude missions in orbits similar to ELFIN's – such as SAMPEX [11], POES [12], and Firebird II [13] – none of which provide either adequate pitch angle resolution or the appropriate energy range in order to study electron precipitation driven by intense whistler-mode waves. The ability of ELFIN to measure precipitating, trapped, and reflected populations of particles in a single spin provided the necessary capability to determine signatures of various types of wave-particle interactions. Every aspect of the ELFIN mission was optimized for this purpose, which therefore led to the implementation of a CubeSat with a highly customized but efficient design.

II. System Overview

An expanded schematic view of ELFIN can be seen in Figure 1. The ELFIN payload consists of three primary instruments: (1) an energetic particle detector for electrons (EPD-E) and an energetic particle detector for ions (EPD-I)

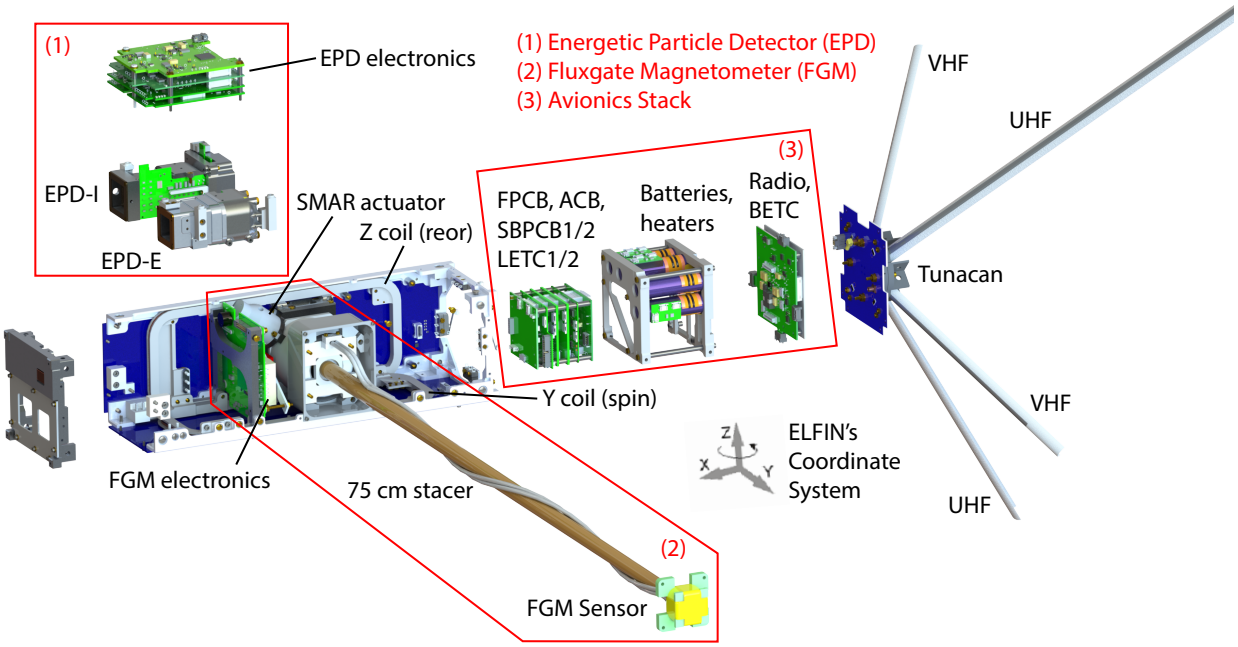

Fig. 1 Interior components of ELFIN are shown in this expanded view. From left to right: (1) the energetic particle detector instrument (EPD) which includes the electronics (SIPS, IDPU, 2 EPD digital boards, preamplifier, and the front end bias supply) along with the electron and ion sensor heads; (2) the fluxgate magnetometer (FGM) sensor mounted at the end of a 75 cm deployable stacer; and (3) the avionics stack which consists of the flight computer, solar battery power boards, attitude control board, radio, and relevant interface boards. The two pairs of deployable bent-dipole antennas are shown on the right, with the two air coils attached to the chassis.

and (2) the fluxgate magnetometer (FGM). The avionics unit is comprised of 8 PCBs and 4 Li-Ion batteries stacked inside of a PEEK (Polyether Ether Ketone, a type of thermoplastic) frame with a thin aluminum and MLI (Multi-Layer Insulation) blanket shielding for electrical and thermal purposes. The flight computer (FPCB), attitude control board (ACB), and two solar-battery boards (SBPCB) were built by the Aerospace Corporation. Each board contains one or two PIC microprocessors with custom software and firmware implemented by UCLA students. Interface and auxiliary boards were also designed and built at UCLA, including two Little Et Cetera boards (LETC1 and LETC2), a big Et Cetera board (BETC), and various small PCBs for battery heaters and harness interconnects. The radio is a custom form factor Helium-82 radio from AstroDev – which is a slightly smaller version of the Helium-100 – capable of VHF uplink and UHF downlink. Power was generated via 20 body-mounted Spectrolab UTJ cells on custom solar panels and stored in 4 Molicel ICR18650J Li-Ion batteries.

There are two deployables on ELFIN: the antennas and the stacer boom. The antennas are stowed in the bonus volume of the 3U+ form factor, and consist of custom rolled up BeCu/fiberglass elements (built by Loadpath) held down by spectraline as shown in Figure 2e. Deployment occurs when a series of redundant burn resistors are energized, which heat and melt the thin spectraline and allow the antennas to unfurl. The stacer boom is a miniaturized version of the axial booms that flew on THEMIS (designed and manufactured by Kaleva Design) and is pictured at the center of

Figure 2a. The chassis and mechanical structure were custom designed and machined in-house. Most components were manufactured at UCLA with 6061 aluminum or PEEK (although brass, copper, tantalum, delrin, and Windform were machined for some applications). The magnetotorquers were custom PEEK frame air coils wound with Elektrisola high-tensile-strength copper-clad aluminum (HTCCA) magnet wire, seen in Figure 2b.

High-fidelity thermal simulations were performed in Thermal Desktop and validated by tests in our Thermal Vacuum (TVAC) chamber. These simulations informed the placement of various thermal treatments and blankets employed throughout the spacecraft. Custom MLI blankets, silver telfon, black kapton, and innovative use of PEEK brackets can be readily seen throughout Figure 2.

Besides the standard challenges of miniaturizing three instruments, electronics, and deployables into a CubeSat form factor, ELFIN's unique science goals imposed unique challenges. By spinning at ~21 RPM with the spin plane aligned with the orbital plane to within 20°, ELFIN needed a custom ADCS solution, communication design, and power strategy. Due to the sensitive fluxgate magnetometer onboard, the spacecraft had to be designed to be magnetically clean, limiting choices on materials, fabrication methods, and test practices. Finally, the spacecraft was designed, built, and operated almost entirely by over 350 undergraduate students across the entire 9 year ELFIN mission (5 year development period, 4 years on-orbit operations), where retaining in-house knowledge and enforcing high technical standards was a significant management-level challenge. There could not be an expectation of full-time commitment, and the development, testing, and satellite operations necessarily revolved around the academic calendar. With students leading subsystems and the entire team, ELFIN's success relied heavily on significant extracurricular student volunteer time.

The challenges associated with a lack of experience within the student-led team were compounded with the complexity of ELFIN's bespoke design, so the development philosophy was to preferentially perform verification by test, rather than analytically. We attribute the nearly flawless performance of the hardware for all four years on orbit to the rigorous testing that various ELFIN prototypes underwent in addition to support from the UCLA staff engineers and technical support from the Aerospace Corporation. A flow chart detailing the Engineering and Flight Model (EM and FM) tests leading up to Pre-Ship Review (PSR) is illustrated in Figure 3, and some of the most important risk reduction tests are listed below.

1) Frequent TVAC tests at component, subsystem, and system levels weeded out electrical and thermal problems early in the design process.
2) Four vibration test campaigns using mostly mass models were performed prior to EM1 in order to identify problem areas that which could be fixed before actual hardware was put at risk.
3) Deployment tests for antennas and stacers were performed multiple times at expected thermal extremes, in vacuum, and after long stowage periods.
4) Range tests, where the UCLA ground station was used to communicate with an RF mockup far away enough to emulate the path loss associated with transmission to space, were performed to ensure that we could close the link.

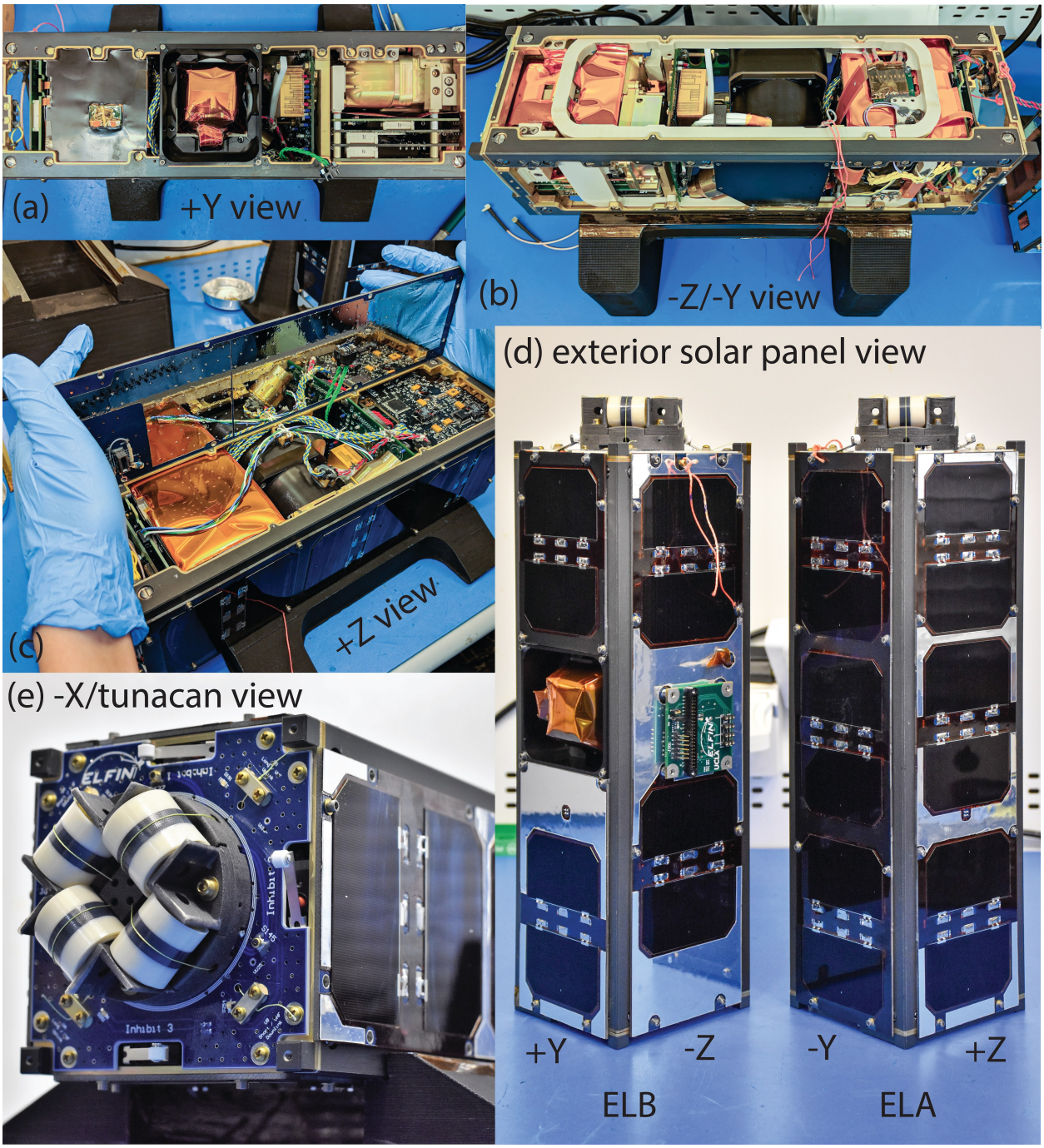

Fig. 2 ELFIN's flight model from various perspectives using axes defined in Figure 1.

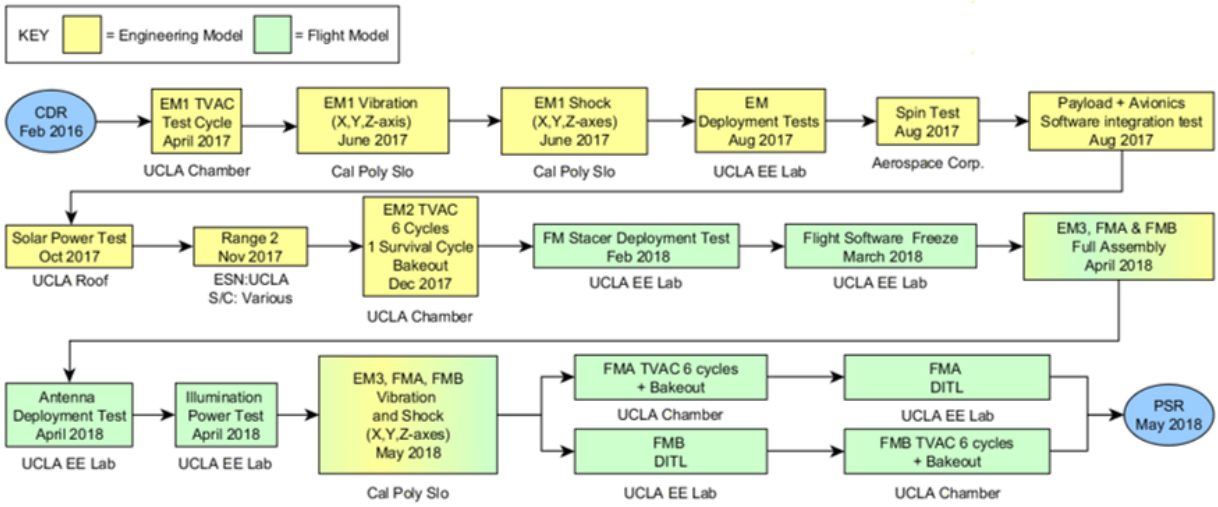

Fig. 3 This shows ELFIN's major tests performed on the Engineering and Flight Model (EM and FM) ELFIN units between Critical Design Review (CDR) and Pre-Ship Review (PSR) and does not include the multitude of testing performed on Development Models (DM) 1-4, RF (Radio Frequency) models, thermal models, and mass models.

There were three Range test campaigns: Range 1 was a pathfinder and resolved exposed technical and logistical challenges; Range 2 successfully showed that we could close the downlink link budget; Range 3 (performed after PSR) showed that we could not close the link on uplink. As a result, the team rapidly constructed a new VHF antenna tower prior to launch, without which ELFIN would not have received ground commands post-launch.

5) Spin tests performed at Aerospace Corporation facilities, where an air-bearing spin platform inside a Helmholtz coil was used to validate spin control functionality.
6) EPD calibration tests with radiation sources (performed after PSR) were performed in vacuum to calibrate the energy bins on both the EPDE and the EPDI in the final flight configuration. These tests showed that all detectors were fully functional and allowed us to precisely determine the preamplifier gain as a function of energy.

III. Operational Overview

As challenging as it was to deliver the flight-ready ELFIN CubeSats on time, we were surprised to find how much more challenging it was to continuously operate a satellite for years while maintaining high data throughput. ELFIN data has enabled a wide range of study primarily due to its novel data products, multi-year statistics, full MLT coverage, and thousands of radiation belt crossings. None of this would be possible without smooth daily operations, even with a fully functioning satellite on orbit.

A good metric for assessing the data throughput (and, therefore, operational success) of the mission is the number of science zones (SZs) ELFIN has successfully downlinked, as itemized throughout various phases of ELFIN's on-orbit life in Table 1. This table paints a clear picture of how challenging the first two years were, even after commissioning. The

Operational Period	Time Range	Total RB Crossings	SZs/Month
Commissioning	Sept 2018 - March 2019	0	0
Calibration + Ops 1.0	April 2019 - June 2020	675	~40
Refactor 2.0	June 2020 - Feb 2021	3255	~360
Inner Belt Observations	Mar 2021 - Nov 2021	5436	~600
Wokring Attitude Control	Dec 2021 - May 2022	2558	~380
Calibrated Ions	June 2022 - Sept 2022	481	~120
		Total: ~12,500	

Table 1 This table shows the various phases within ELFIN's on-orbit operations. The mission reached a turning point in June 2020 with the release of Refactor 2.0, which greatly enhanced ELFIN's scientific feasibility. Outer Radiation Belt crossings are from L $\in [3, 18]$, while Inner Radiation Belt data collections span L $\in [1.1, 18]$. Only observations where data completeness $> 60\%$ over the whole L-shell range is counted and significantly more data exists for smaller fractions of SZs.

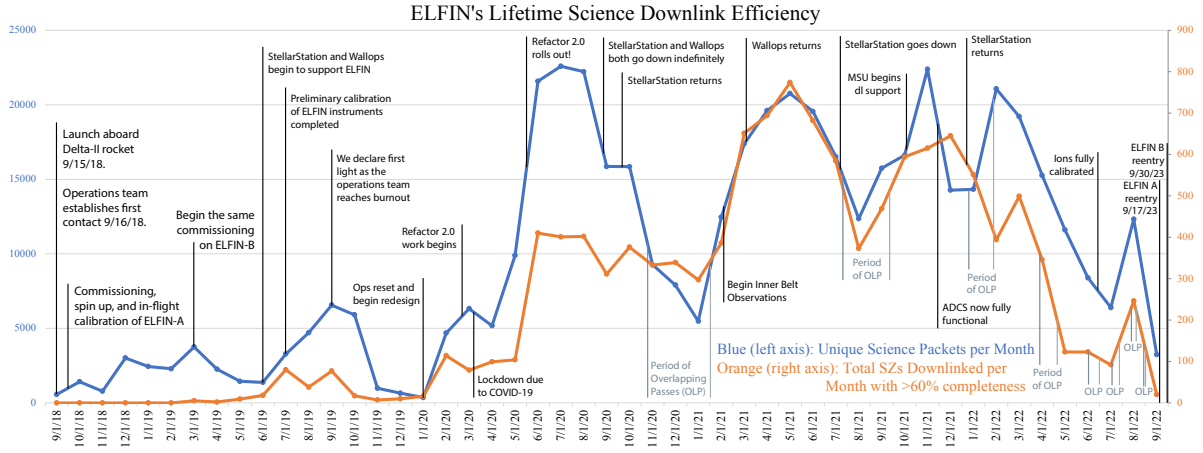

Fig. 4 We show ELFIN's science packet downlink efficiency (blue, left axis), measured in science packets downlinked per month, as well as science zone downlink efficiency (orange, right axis), measured in number of radiation belt crossings with greater than 60% completeness ratio (ratio of downlinked to collected data packets) per month. We annotate various phases of the mission here, showing the large impact of Refactor 2.0 and how it turned ELFIN into a viable science mission. Various effects cause the efficiency to vary throughout the mission, such as variable external downlink support and increased downlink volume required to support more science/operational objectives.

$> 12,500$ radiation belt crossings downlinked by ELFIN would not have been possible without a complete rethinking of the operational paradigm, called Refactor 2.0, that was implemented in June 2020. This is shown in Figure 4, where ELFIN science zone downlink rates improved considerably after the refactor and remained sustainable until the end of the mission. Science zone downlink efficiency after Refactor 2.0 fluctuated as a function of partner downlink station availability (partners included NASA's Near-Earth Network (NEN) dish-antenna on Wallops Island, StellarStation's antennas in Tokyo, and Montana State University's UHF downlink station), downlink complications from ELFIN overlapping passes (or OLPs, when both ELFINs are overhead during the same passes, effectively halving the available pass time), and desired data volume (SZs/month decreased when downlinking additional inner belt data, attitude data, and/or ion data corresponding with each of ELFIN's more advanced operational phases). It took several months to optimize our operations paradigm around the new software, and our first major update was released shortly before the first set of overlapping passes (which included significantly more optimized science packets) occurred at the end of 2020. This significantly reduced the error rate at which we downlinked data, as exemplified by the sizable drop in the blue curve (the total number of science packets downlinked), while the orange curve (the total number of SZs downlinked with completeness $> 60\%$) remained relatively stable. When ADCS became fully operational in November 2021 – demonstrating attitude determination and control of both ELFINs to $< 1^\circ$ of precision – it significantly increased the quality of ELFIN science data at the cost of reduced overall science data due to the increased attitude data downlink requirements and even more constrained spacecraft resources. Combined with ion data becoming fully online in June 2022 – where each science zone downlink volume now increased by 50% – and the rapidly increasing OLPs, the number of SZs downlinked decreased towards the end of the mission. The lower SZ coverage was worth the added value of fully contextualized ELFIN data with all instruments working.

The paper is organized as follows: Section IV will cover the implementation of flight software and avionics onboard ELFIN. Instrumentation and other mission-level details are already covered in Angelopoulos et al. [6] and will not be repeated here; Section V will cover the technical implementation of Refactor 2.0 and present lessons learned; and finally, Section VI highlights some of the scientific contributions enabled by ELFIN.

IV. Technical Overview

A. Avionics

FC+WD Paradigm ELFIN's high-level avionics system block diagram is shown in Figure 5. There are two main UART (Universal Asynchronous Receiver-Transmitter) lines on the avionics stack: UART1 handled all intra-spacecraft communications, while UART2 handled all external communications via the radio. There are two PIC microprocessors on the FPCB: the main Flight Computer (FC), a PIC18F8722, and an external Watchdog (WD), a PIC18F6722. The FC is the central hub for the whole spacecraft and its main responsibilities were as follows:

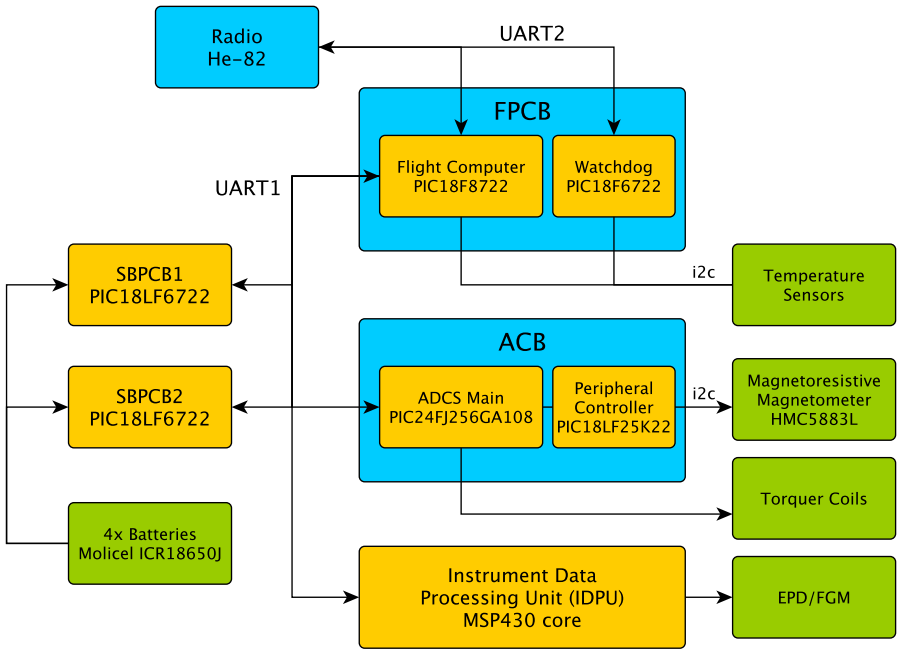

Fig. 5 Digital block diagram showing ELFIN's two primary UART (Universal Asynchronous Receiver-Transmitter) lines. There are six PIC microcontrollers that work in tandem to maintain spacecraft power, telemetry, telecommanding, and attitude control.

1) Parse, authenticate, and execute commands received via UART2 from the radio
2) Send commands to the radio and send beacons at specified regular time intervals
3) Command and interface with other boards on the spacecraft (2x SBPCBs, an ACB, and an IDPU)
4) Host script and scheduler functionality and keep track of spacecraft time
5) Aggregate and store spacecraft housekeeping data
6) Maintain logic for safe mode entry and error logging
7) Control core functionality throughout the spacecraft, such as deployables, heaters, and instruments
8) Regularly check aliveness of the WD

The WD – not to be confused with the PIC's internal watchdog – serves as a more robust, programmable, and external watchdog. It sends/receives heartbeat pulses to the FC and resets the whole spacecraft bus if it detects the FC is unresponsive (conversely, if the WD is unresponsive, the FC will reset only the WD, as they are each on separate power rails). The WD resets the FC and main bus on an aggressive time-scale – once every 2 hours – to avoid latchups or undesired effects from extended runtimes on our main program loops. Having both FC and WD on the same UART2 line connected to the radio meant that even if the FC became unresponsive, the WD could parse commands and thus could attempt FC recovery procedures.

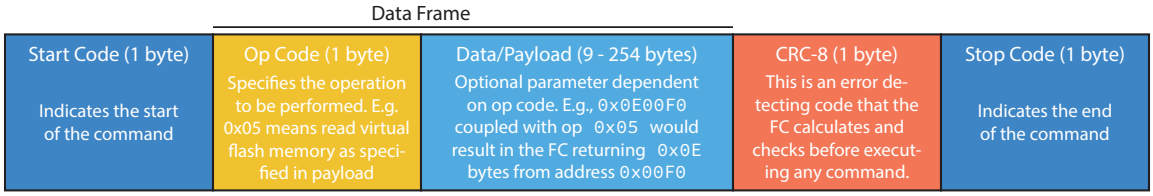

Fig. 6 ELFIN's command protocol

Command/Data Protocol The avionics stack runs on our own custom software and firmware across six PIC microcontrollers. Of these, the FC is the brains of the system, managing TT&C, as well as the scripts and scheduler. All of this is based on a simple command and data protocol which, minus authentication, contains a 1-byte start code, a 1-byte op code, a variable-length command payload, a 1-byte CRC (cyclic redundancy check), and a 1-byte stop code (Figure 6). The start and stop codes signify the beginning and end of a command, while the op code is a single byte that maps to a particular activity that can be executed by its corresponding PIC. The command payload would be the parameters (if any) for the op code; for example, a payload of `0x0E00F0` following a `0x05` (the READ op code) would make the FC read and return 240 (`0xF0`) bytes read from address `0x0E00`.

Packet sizes are limited to the payload limit of the AX.25 packet, which is 256 bytes. If special codes (start, stop, or escape codes) appear in the payload, they would be escaped using the escape code. This could grow the size of packets by a few bytes, so in general, we would limit our command payload size to 246 bytes.

Although this limitation would govern uplink efficiency, most individual commands are much smaller than this maximum size. We take advantage of this by stuffing multiple ELFIN commands into a single AX.25 radio packet. When ELFIN receives a radio packet with multiple ELFIN commands, the FC's command handler processes each of them sequentially. Because radio packets are processed atomically by the FC, commands that are bundled together are guaranteed to also be executed together. This is how we acknowledge (ack) receipt of commands on ELFIN, for example: a WRITE command is accompanied with a READ command, which reads back what was just written such that it can be validated on the ground. In the original operations paradigm, this was validated by eye in real time during passes, but after Refactor 2.0, this was automatically compared to the desired intent of the command, and programmatically verified.

Despite our ack implementation necessarily meaning the successful uplink of the associated command, the inherent communications design between the MOC and ELFIN's FC still remains an unreliable service; that is, it does not notify the user if delivery fails. For example, sending a WRITE command to the spacecraft could be successful, but on the ground, it is possible to not receive the ack (e.g. due to a weak RF link or an issue in the downlink pipeline). Because of that, we can no longer be sure of what is onboard the spacecraft, and without the right software, operators can easily lose track of spacecraft state. This uncertainty would pose major challenges to achieving error-free operations, and, therefore, more stringent precautions were taken to ensure consistent knowledge of spacecraft state in Refactor 2.0.

Authentication At the time of ELFIN's design, NASA CubeSat missions did not require uplink authentication or encryption. However, we wanted to ensure that our spacecraft was resilient to replay and DoS (Denial-of-Service)/spam attacks, so we began by implementing a SHA-1 (secure hash algorithm 1) hash, calculated using an incrementing salt, in every uplinked command. However, the FC's PIC18FLF8722 was not powerful enough, taking multiple seconds to calculate the hash onboard, forcing us to look for alternative solutions. In the end, we implemented a version of the lightweight SipHash function [14], modified to operate on 32-bit values.

$$\text{MAC}\,(P, S, K) = \text{Hash}\,(K, \text{Hash}\,(K, P \parallel S) \parallel S) \tag{1}$$

For every uplinked command, a 64-bit hashed message authentication code (MAC) is appended to each packet, as formulated in (1), right before the CRC and stop code. This code is calculated by taking the byte payload and appending a 4-byte salt, which is hashed with a key. The 8-byte hash result is again hashed by a second key, to produce an 8-byte tag that can be used to authenticate our commands and is resistant to length-extension attacks.

The original idea was to have the 4-byte salt pseudo-randomly change as a function of RTCC, command counters, and various other hskp statistics. However, due to a push for simplicity and a fear that we would lose the ability to command the satellite in the event we lose sync with the clock (or lose the clock altogether), we chose to manually set the 4-byte salt and private key and regularly change them. The salt is publicly broadcasted in our beacon, and in the event ELFIN receives a command containing a bad hash, it will respond with an 8-byte sync MAC concatenated with 2 bytes of the salt, shown in (2).

$$\text{Sync Mac}(S, K) = \text{MAC}(\text{Sync Message}, S, K) \parallel S_1 S_2 \tag{2}$$

Due to the simplicity of the SipHash algorithm, we can quickly brute force the resynchronization in < 10 s using this bad hash response (this feature is toggleable, since leaving it on makes ELFIN vulnerable to DoS/spam attacks).

The other layer of authentication simply relied on the standard AX.25 callsign, which is a part of the AX.25 packet structure and therefore comes for free without costing packet payload size. This feature is included in many modern amateur radios, but not on our Astrodev Radio. As a result, we had to add callsign filtering functionality as an over-the-air (OTA) update, as discussed over the next few paragraphs. These two simple authentication strategies incur little computational cost. Thanks to their simplicity, effectiveness, and successful implementation on ELFIN, NASA adopted authentication requirements into the Space System Protection standard (NASA-STD-1006) for CubeSats and SmallSats without propulsion.

Reprogrammability In-flight reprogrammability was a major risk reduction feature that was very rudimentarily implemented late in our development cycle. However, it proved to be incredibly useful both before and after launch.

Over-the-air (OTA) reprogramming involved directly writing assembly instructions into program flash via UART commanding. This would be done by first setting several flags at disparate addresses which then allow the FC to directly write to its own program flash one page at a time. Our reprogrammability implementation on ELFIN had no fail-safes, and as such, was only used when no other options were viable. To make things easier, various configuration settings, default values, and other program constants were deliberately stored at fixed locations in program flash. Reprogramming something more complicated would often involve writing the new instruction set into a region of unused program memory, then strategically inserting jump instructions without altering other code. This strategy relied on the fact that the MPLAB PIC compiler was not perfectly efficient, and we could often find optimizations at the assembly level which would provide enough space to insert a jump instruction which would point at a newly injected set of assembly instructions written into empty program space.

During development, we had tested in-place single page reprogramming, multi-page writes (up to 64 pages, one page at a time), and jump insertions. Importantly, we relied on detailed procedures and pre-tested scripts to ensure spacecraft safety during reprogramming events. Our first opportunity to verify this was after ELFIN was already fully built for flight and shortly before delivery. We had realized that there were transients in the PIC GPIOs that could potentially lead to inadvertent early deployments and opted to surgically inject new code to rule out that potential fault path. In addition, we discovered how the Helium radio parsed and transmitted callsigns (it was not in their documentation), so we added a jump to a new callsign filtering function within the execution handler. This made ELFIN more resilient by allowing it to respond only to commands with the appropriate TX/RX callsigns. These were both performed on ELFIN A and ELFIN B after they had already been configured for flight, demonstrating how powerful this capability is while also practicing this capability should the need arise in flight.

Fortunately, an in-flight OTA reprogram occurred only once in flight. A few months after launch, the operations team needed to change the battery heater setpoints on the temperature sensors (TMPs) as the orbits were drifting into longer and colder eclipses. Our WD was originally responsible for initializing and controlling this thermostat behavior, but unfortunately, the communication between the WD and TMPs was found to be unreliable shortly after launch on both satellites. The root cause of this was never determined, but the primary FC-reset functions of the WD were unimpeded. However, we needed reliable TMP sensors and, because of the redundant control of the TMPs built into the hardware, the FC could actually be reprogrammed to initialize and write commands to the TMP sensors. As a result, we wrote new functions and op codes for FC-TMP initialization and setting configurations. This entire workflow was successfully implemented on both ELFIN satellites 3 months after launch, and led to full confidence in ELFIN's batteries remaining within thermal limits.

Scripts and Scheduling Similar to chaining multiple commands in a radio packet, series of commands can be stored in flash and called upon to be executed altogether. This is called a script, where commands are sequentially stored in

flash and are no longer limited by the byte-size limit of radio packets. The primary difference here is that the command sequences are bookended by script start and script stop commands. Scripts are identified by their stored address, and the FC can be commanded to execute any valid script (defined by containing valid commands (which must have valid op codes, payloads, CRCs bookended by start/stop codes) bookended by start and stop script commands). There is a unique script, the boot script, which is a script that is run on first boot. This is useful because scripts are flexible and can be easily modified in operations; throughout the mission, the boot script has been updated to change how successful boot has been signaled, configure battery usage, set particular battery heater setpoints (after the successful OTA reprogram mentioned earlier), or modify housekeeping configurations. It would have been helpful to implement these "checkpoint" scripts in more locations (i.e. every time instruments turn on, or whenever safe mode is entered) as these would allow for easy and safe modifications of onboard behavior via just scripting. To prevent infinite loops, scripts are unable to execute other scripts, but scripts are, instead, allowed to modify the scheduler and, in turn, the scheduler is designed to execute scripts at designated times.

The schedule is a region of flash space that consists of 256 8-byte entries. Each entry is made up of a 6-byte timestamp in Binary-Coded Decimal (BCD) format followed by a 2-byte address which points to a script. Bit flags within the 8-byte schedule entries in the most significant bits for the hour and minute bytes are used to flag the status of the schedule entry, ranging from live (unmodified), executed (MSB on hours), or stale (MSB on minutes). When the FC comes across a timestamp from a live entry that is greater than the current time, the FC will mark the entry as executed and then execute the script at the address of the associated schedule entry. If the FC finds an entry that is scheduled for a time later than one minute after the current time, the FC will mark the entry as stale and will not execute its associated script. This prevents scripts from inadvertently going off at unintended times should a schedule entry fail for any reason (such as browning out from low battery or being busy with other tasks). The FC reads through the scheduler on each wake-up cycle, around two seconds, and thus, the precision of the scheduler is <2 seconds.

Scripts can schedule or reschedule new entries, and there are a 3 ways this can be done: reschedule to a new specified time, reschedule relative to the current time, and reschedule relative to the existing scheduler time. Rescheduling scripts allows for recursive events to occur (and were intentionally used on various occasions), but is not a flight risk because (1) we can disable the scheduler by entering safe mode and (2) the FC checks for commands after running a script.

Advanced functionality relied entirely on this interplay of using scripts to reschedule other scripts. It would not be until halfway through operations – when Refactor 2.0 was implemented – that we began to maximize the potential of this paradigm. An early example of the effectiveness of this simple design was our payload power ON sequence. This sequence is actually a series of a ~300 precisely timed commands stored across 8 different scripts totaling a few kilobytes and is summarized below:

1) The first script initialized several other scripts using reschedule commands and is the only script address that will show up in the scheduler (i.e. one science collection takes only one schedule entry). This script had multiple

responsibilities: (1) it set the science collection length by rescheduling the stop collection/payload off script; (2) it selected the script that booted the desired instrument program; and (3) it rescheduled the payload power initialization script for two seconds later (i.e. to be executed on the next wakeup cycle).

2) The power initialization script would immediately run on the next FC wakeup cycle, turning on the payload power rails one by one, in a safe and controlled manner. If stable, the last command in this script would reschedule the next script, a specific EPD configuration script, to be executed. Our reschedule command functionality had the capability of rescheduling without changing the associated script address, which meant we had fixed schedule entries that served essentially as a particular configuration saved in flash. This means that choosing the desired EPD configuration is as simple as writing the two bytes associated with the address of the desired EPD configuration script into a fixed schedule entry.
3) The EPD configuration script – which configures energy bins, histogram settings, and coincidence logic – ends by rescheduling the sector configuration script, which is another fixed schedule entry.
4) The sector configuration script would have specific configurations for how sectoring would work (e.g. zero crossing determination algorithm, phase offsets, and other sectoring settings). The last command here would finally schedule the actual science recording to begin.
5) The stop collection, scheduled by the first script at another fixed schedule entry location, would finally be executed, ending the collection and shutting power down safely.

In this example, several scripts – which handled simultaneous payload housekeeping and attitude data collection – were left out to keep the example straightforward. This fairly simple software setup resulted in an easily adjustable way to schedule fairly complex science operations using just a single schedule entry. However, this hidden complexity was really a downside in disguise and hindered our operations as time progressed due to software that became progressively more challenging and unintuitive to use as the number of activities increased. It was difficult for operators to reliably keep track of both ground and spacecraft state, resulting in a large number of human errors and very poor operational efficiency. This problem was rectified with the complete operational overhaul, Refactor 2.0, described in the following section.

V. Refactor 2.0

In January 2020, we began working to identify flaws with our original operational paradigm while designing a new paradigm that would streamline satellite operations. In March 2020, a tiger team of undergraduates was assembled, and after a technical peer review, began intently working on what would ultimately be called Refactor 2.0. This team included Akhil Palla, Jason Mao, Sharvani Jha, Austin Norris, Chanel Young, James King, and myself. The release of Refactor 2.0 on June 12, 2020 immediately resulted in operators spending 6x less time doing operations (from 3 hrs every day to 1 hr every other day) while successfully downlinking nearly 10x more science zones per month

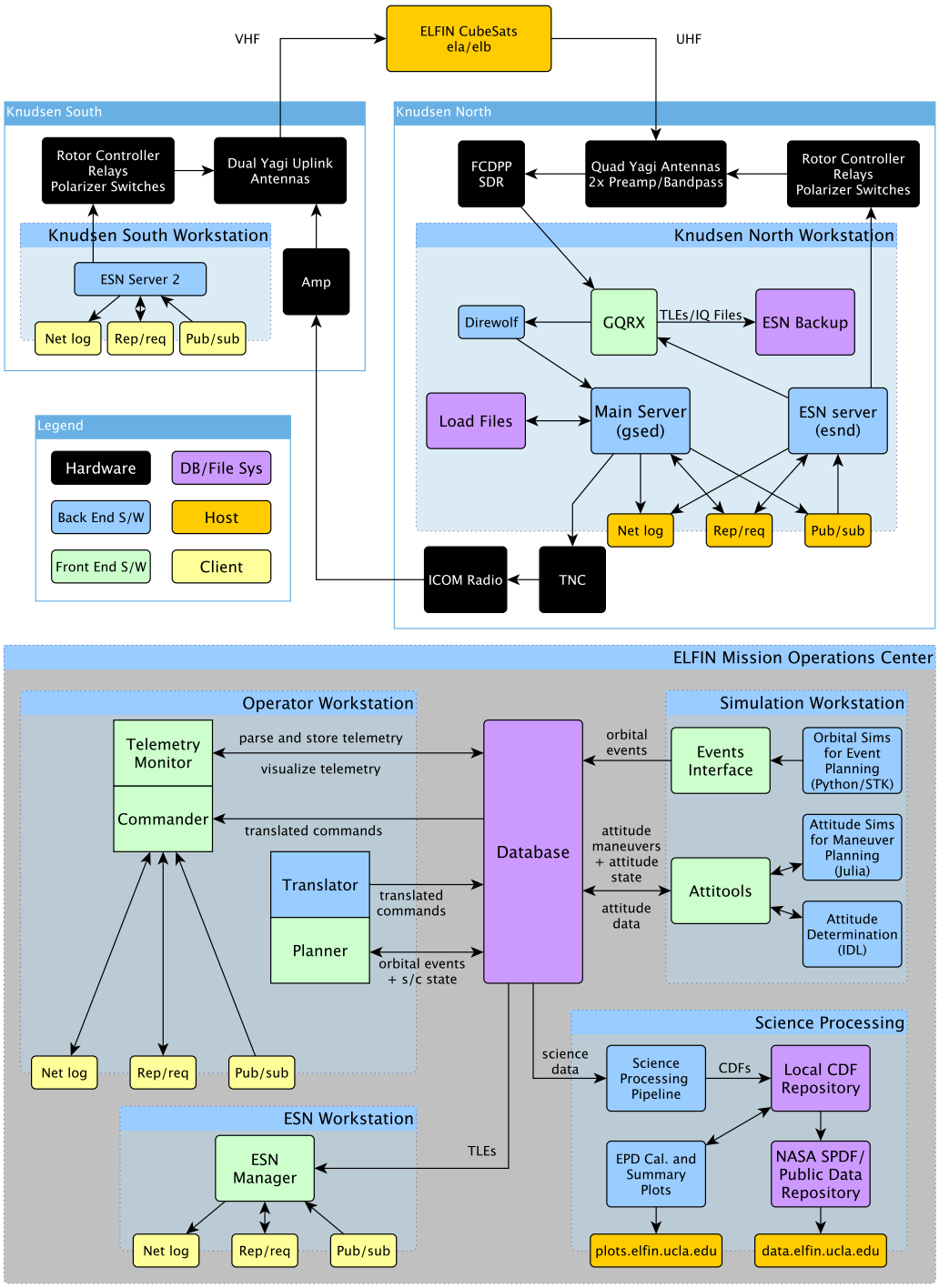

Fig. 7 Block diagram showing all the various ground components necessary for daily ELFIN operations. The student operations team interfaces primarily with software in the MOC workstations, particularly via the planner and commander. The student developers are responsible for software that handles pass management, satellite tracking, and real time commanding, as well as orbit/attitude simulations, operational planning, science processing, and data distribution.

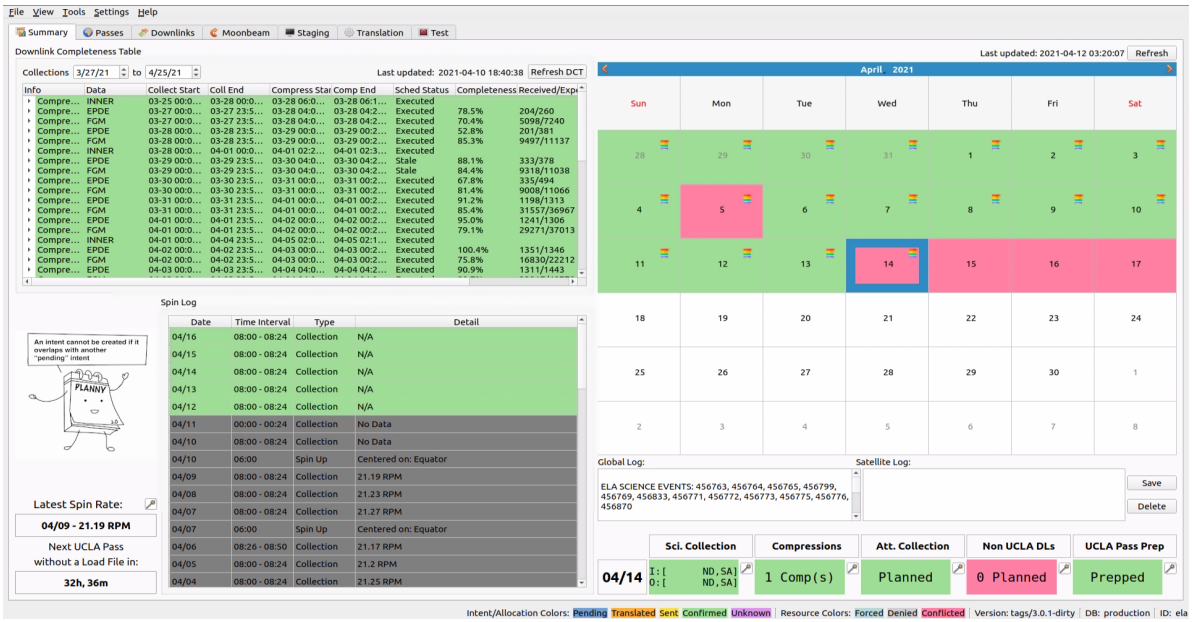

Fig. 8 A screenshot of the Planner in April 2021. In the calendar view, green denotes when all necessary Intents are confirmed aboard the spacecraft for that day. This allows operators to quickly tell which days and what activities have yet to be planned. Intent/Allocation states are color coded on the bottom bar. Clicking on the wands will bring up Intent creation wizards for each type of Intent. Clicking on each day shows a summary of the Intents planned at the bottom right. The 14th is red because Downlinks Intents over non-UCLA stations have yet to be created. The Downlink Completeness Table (DCT) on the top left side provides up-to-date downlink completeness percentages and Intent states for each Science Compression Intent. Expanding each of those would show the associated downlink completeness and Allocation state for each of the associated science collections. The spin log on the lower left corner shows the latest spin rates.

(from ~40 SZs/month to ~360 SZs/month). This was achieved by rewriting all ground software and performing minor modifications to onboard FC behavior in order to conform to a newly abstracted structure. In a departure from previous paradigms, the new workflow sought to leave behind ground flexibility in pursuit of ground truth wherever possible. The graphical user interface (GUI) would also be completely redesigned to provide useful state information in a transparent and contextual manner. The following subsections will introduce and define the requisite operational abstractions before explaining the rules and core philosophy.

A. Key Concepts

The original operational design for planning and scheduling of spacecraft activities was accomplished in the **Planner** software module (running on Operator Workstations in Figure 7). The original Planner had a timeline view which allowed the user to easily view all pertinent orbital events, and add spacecraft activities anywhere and anytime. However, matching all these activities with resources used on the spacecraft was too complex to automate, and was therefore done manually. As a result, this flexibility led to a confusing experience, where operators were oftentimes juggling several tens of different activities daily, manually keeping track of how the state of each activity evolved over time, and figuring out how the activities eventually mapped to memory to determine where future activities could be stored. As a result, the operations team had to maintain several supplemental spreadsheets just to assist with the daily operations of ELFIN.

The refactored Planner (Figure 8) did away with the timeline view, instead opting to build the new workflow around

"wizards" (similar to generic software installation wizards) where the user interface would guide operators through a sequence of small steps, allowing them to plan each spacecraft activity while following rigid rules. Each spacecraft activity would be associated with an **Intent** – which refers to a high level spacecraft activity, such as collecting science, collecting attitude data, performing attitude maneuvers, compressing science data, etc. – thus, each Intent would have its own dedicated wizard workflow. Each Intent contains a **Scope**, which is the property that allows Intents to be compared to each other, thus allowing the software to ensure that Intents do not overlap or conflict. For example, science zones were scheduled on a daily basis, so the scope of a Science Collection Intent would be just the date itself. A single Science Collection Intent contains the instructions for all science zone collections for that particular day and the Planner would be responsible for ensuring only one Science Collection Intent for any given date can exist onboard the spacecraft.

To accomplish the activity associated with an Intent, there must be some utilization of physical resources (i.e., memory), either for schedule entries or script space. We therefore discretize the physical memory into **Resources** and define an abstraction called **Allocations**, which refer to the mapping of Intents to their respective Resources. For example, a Science Collection Intent which scheduled 12 science zones in a day would require 12 schedule entries. The Resource for a Science Collection Intent is defined as a chunk of 4 schedule entries (a predefined 32 bytes address space, since each schedule entry is 8 bytes), so there would need to be three Resources allocated for this particular Intent. The Allocation therefore refers to the association of these three Resources with this particular Intent and are thus the most important abstraction to track. Before the refactor, a significant fraction of operational errors involved inadvertent overwrites in memory. To prevent that, the Planner and Commander would enforce rules that disallowed partial modification of any component of Intents. Intents, Allocations, and its associated uplink commands must all be atomic in order to preserve the integrity of our assumptions made when tracking state. In the earlier example, we would still need three resources even if we were collecting 9 science zones in a particular Intent, preferring to often have unused schedule entries in favor of accountable resources.

Even with constraints and rules in place, operations often change, and the software must be flexible enough to change an Intent even after the Allocation has made it onto the spacecraft. This can be accomplished using **Wipes**, which are essentially blanked Allocations of equivalent size for a particular Intent. Atomicity is therefore preserved during Intent modifications. This prevented the existence of partial Allocations onboard the spacecraft, an unfortunately common occurrence before Refactor 2.0 which often led to non-deterministic behavior from the perspective of ground operations.

Once Allocations are created within the Planner, they are translated into commands via the **Translator** – a software module within the Planner – and exported to the database. Operators can then use the **Commander** to convert the translated output into a **load file**. The Commander is a separate software module that manages both real-time commanding and Load Files: files that store series of commands and associate them with a specific pass. These Load Files are saved on the Main Server and can be executed autonomously even in the event of communication loss to the MOC. See Figure 7 for further details.

With these abstractions in place, we now define the data and knowledge models that then allow for the specification of rules which govern Intents and Allocations. We first begin with the **Ground Truth Model**: an idealized representation of state assuming a perfect computer network with no risk of data loss. It splits the classification of Allocations into three categorizations:

- **Pending** refers to Allocations that do not yet exist aboard the spacecraft (i.e. they exist only in the Planner).
- **Live** refers to Allocations that have been written to the spacecraft and are therefore going to be executed.
- **Stale** refers to Allocations that that have been overwritten or are otherwise invalidated due to, for example, an elapsed timestamp.

However, our communications link is not perfectly reliable (it is possible to drop uplink or downlink packets due to RF nulls (from spinning) in our communications) and our infrastructure within a primarily educational institution – rather than a primarily space mission facility – does not guarantee 100% uptime for both network access and power, meaning that our knowledge models must be resilient against failures in both uplink and downlink: we cannot always assume commands will successfully make it onto the spacecraft. In order to handle this, the Planner's state tracking relies on the **Planner Model**, which slightly expands upon the Ground Truth Model:

- **Pending** now refers to Allocations that have not yet been translated into a command (i.e. they are still within the Planner).
- **Translated** refers to Allocations that have been translated but not yet transmitted (i.e. they have left the Planner and are potentially in a Load File waiting to be uplinked during an upcoming pass).
- **Sent** refers to Allocations that have been sent to the spacecraft and may exist onboard, although they have not been verified yet.
- **Confirmed** refers to Allocations that have been verified to be written aboard the spacecraft.
- **Stale** refers to Allocations that have been overwritten by another "confirmed" Allocation or are otherwise invalidated by an elapsed timestamp.

Verifying commands is done by the Commander in real time upon sending/receiving each command. Each command contains both a write and read bundled into the same command (see Section IV.A) and contains both an associated Intent ID and associated Allocation ID. If the readback of the onboard Resources onboard match expectations, the Allocations are marked Confirmed in real time. This is important because in special cases (e.g. wipes), it is desirable to only uplink commands if previous commands have been confirmed successful. An Intent state will reflect the weakest state of its Allocations, so that an Intent will only be confirmed if all Allocations within it are also confirmed. These states are color-coded in the Planner GUI (see Figure 8) at both the Intent level (for the overall summary view) and Allocation level (when using the wizard).

The **constraint checker** is the underlying software running within the Planner, that handles logic and adherence to Refactor 2.0 rules throughout all the wizards and planner actions. The states from the Planner Model are further

grouped into ground truth states which is used for constraint checking:

- **Possibly live:** "sent" and confirmed".
- **Possibly stale:** "stale" and all but the most recent from "possibly live".
- **Future live:** "translated", "sent", and "confirmed".
- **Future stale:** "stale" and all but the most recent from "future live".

The logical flowchart shown in Figure 9 shows how the constraint checker enforces these rules and acts as a backend module within the Planner which other modules can interface with.

B. Rules, Assumptions and Overall Philosophy

Leveraging the framework described above, the constraint checker can now enforce the two directives that govern safe operating:

1) Prevent any scheduling conflicts between activities.
2) Handle uncertainty in the safest and strictest way possible.

Prevent scheduling conflicts: Intents with overlapping scopes cannot coexist onboard the spacecraft. If a new Intent overlaps with any existing Intents, the new Intent must include all the resources for which the overlapping Intents have non-stale Allocations that are not wipes. This allows us to make the Intent Scope Assumption: if two Intents of the same type overlap, their scopes are identical. Additionally, new Intents may not interfere with any other already-scheduled activity, and using scope, time-allocation rules must prevent overlap with all other non-stale Intents.

Handle uncertainty: Disallow overwriting active schedule entries and discourage situations in which a new Intent allocates a non-stale resource. If a new Intent allocates a non-stale resource, it must allocate all other non-stale resources which share the same Intent. We also enforce the ordering of Allocations such that any Allocation can only be replaced by a newer Allocation. Stale Allocations are no longer actionable, and an Allocation can only be translated or transmitted if it is not future stale. Because pending Intents have no explicit ordering when they exist within the planner, we enforce stricter rules for pending Allocations. Thus, a resource may not be re-allocated while there exists another pending Allocation, and an Intent may not be created if it overlaps with another pending Intent.

These lead to a few key assumptions. The In Order Assumption: Intents exist onboard the spacecraft in the same order in which they were allocated on the ground (FIFO), barring any Intents which were never successfully written. Only the latest translation output can be included in a load file, and if load files contain multiple translated Allocations for the same resource (i.e. wipes or overwriting of resources), the older Allocation only appears in earlier load files than any Allocations which are newer than it. This required upgrading the Commander to be capable of real-time command verification and additionally have conditional commanding, opting not to send commands until its prerequisite commands are verified to also be onboard.

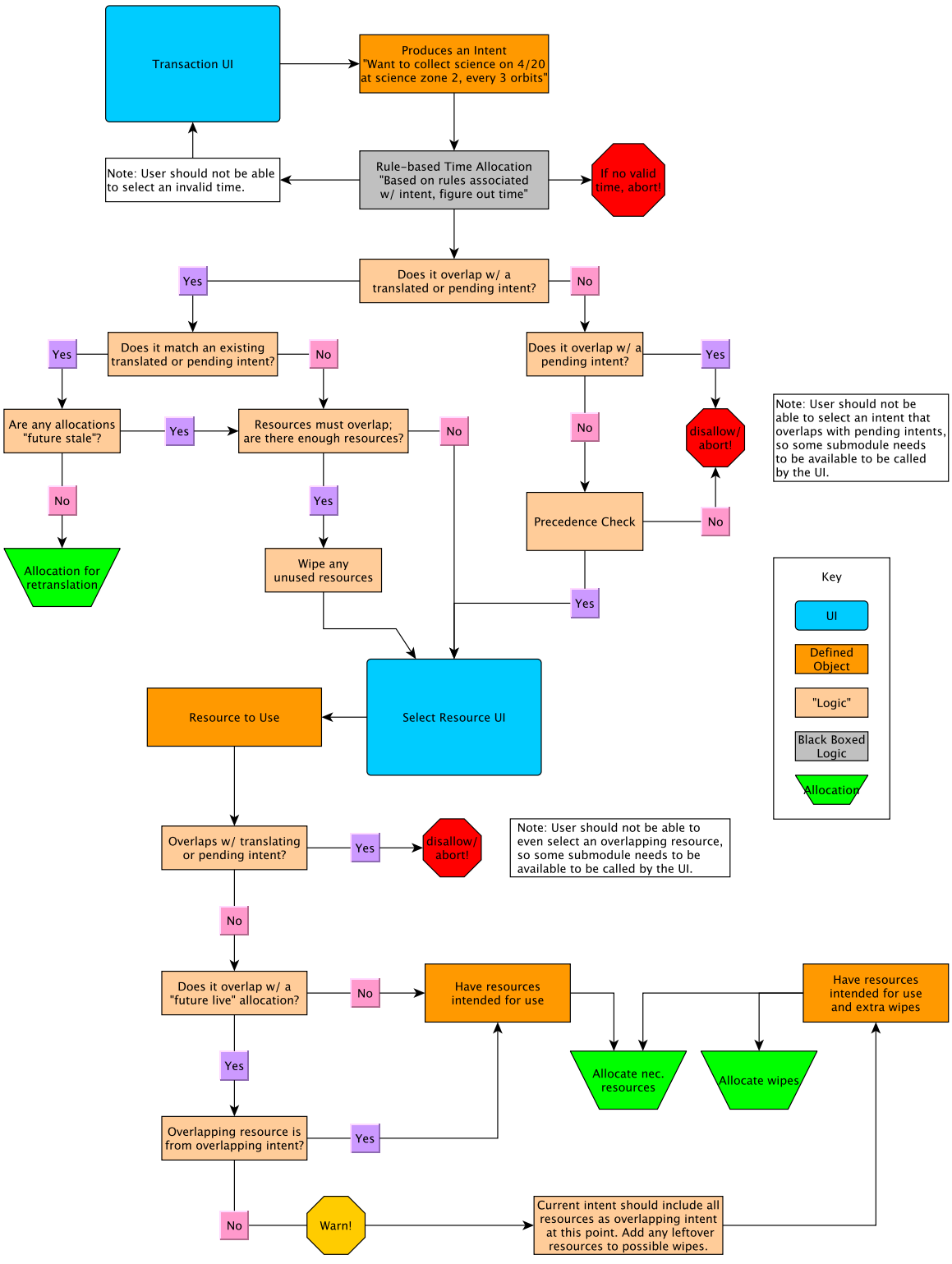

Fig. 9 Logical flowchart demonstrating the type of decisions the constraint checker would make in order for operators to safely make Allocations.

We can also make the atomicity assumptions: Allocations are atomic, and Intents are pseudo-atomic. Intents cannot be fully atomic – due to the byte-size limitation of AX.25 packets – and some Intents are too large to fit in a single radio packet. This requires Allocations to only be transmitted along with the rest of their Intent. The discrepancy between the atomicity of Intents and Allocations means special care must be taken to ensure that a mix of two Intents can never simultaneously exist (no "Frankenstein Intents"). If an Intent has too many Allocations to fit in a single uplink frame, the first frame includes a wipe of the entire resource, and the second frame is only transmitted after the wipe succeeds, relying on the conditional commanding framework alluded to in the previous section.

These rules served to provide structure to the complete operational freedom of the original design that allowed for too much uncertainty. Whenever an option presented itself, the Planner would delegate set choices to the operator in a safe manner, as dictated by the constraint checker, rather than giving the operator free reign. For example, the resource-picking step in each Intent creation wizard would present a list of all resources with color-coded statuses; the operator can then see all resources and deliberately pick resources rather than allowing software to automatically pick any free resource. There was still operational freedom, since the operator could pick a live conflicted resource if they so wished, but the constraint checker would force the selection of all other associated resources to preserve the atomicity rule. When generating commands, the Commander would automatically insert wipes as needed to ensure safe uplink of these Allocations. With thoughtful design, these choices were easy for the operator to make because we chose to keep the software transparent while avoiding hidden complexity and nondeterministic behavior.

C. The Downlink Completeness Table (DCT)

Presenting the operator with contextually useful information was the final major aspect of Refactor 2.0. By taking advantage of the framework described above, we could finally achieve an intuitive and linear operating experience, always providing only the information necessary whenever the operator was presented with decisions. For example, the attitude maneuver wizard would present historical spin rates, historical attitude states, and available/past resources, so operators could make informed decisions when making attitude maneuver intents. However, the most consequential impact from Refactor 2.0 came from the **Downlink Completeness Table** (DCT), a preview of which can be seen in the upper left side of Figure 8. It worked by integrating key aspects of the science data processing pipeline into the Planner and leveraged the new layered levels of insight afforded by the paradigm of Intents and Allocations in order to give operators direct access to information regarding collections of science data, onboard data processing/compression status, and downlink status/completion rates.

The DCT shows all science collections (based on Science Collection Intents), grouped into uncompressed and compressed (based on Science Compression Intents). This was already very helpful because tracking all the different timestamps associated with successfully downlinking science data was very confusing. There are five different types of times that the operator must keep track of when downlinking data that come from each science collection intent:

(1) the science collection intent scope; (2) the *n* timeranges associated with the *n* planned radiation belt crossings throughout that given day; (3) the science compression intent scope; (4) up to four science compression timeranges (one for each product: EPDE, EPDI, FGM, and inner belt data); (5) the downlink time (of which data is typically downlinked over 3+ passes to ensure high completeness percentages). This quickly becomes confusing, as there is a large and growing number of new timestamps required for keeping track of command state and future command generation that are produced daily. The proper timerange for downlinking usually does not match what is naturally spoken and can lead to mistakes and confusion. For example, if a scientist communicates to the operations team that the data from January 1st is incomplete and important, the operator would need to first figure out when the data from Jan 1st was compressed – say for example it was Jan 2nd between 00:00 and 00:30 – then examine the passes associated with downlinking the compressed data to see if there are still future downlinks. If there were issues with the compression, which was common in the earlier operations paradigm, the operator would schedule new compressions of the same data, which would now add even more timestamps to keep track of. Before the DCT, this was tediously tracked using error-prone spreadsheets. The DCT's transparent metrics were contextually presented to the operator throughout the wizard for generating data downlink intents, allowing the operator to make the best decisions in prioritizing which data to downlink without ever having to type in timestamps. The user can click and drag exactly what data is desired, and the DCT/Planner software will automatically determine the correct timerange to request. This intuitive selection process eliminated the risk of making typos, which is common when dealing with so many timestamps.

D. Autohban

The DCT not only saved a lot of time on operations, but also provided a means to rapidly analyze the new influx of science data and radiation belt observations. This was achieved with the Autohban, short for Automated Tohban report, which is a tool for aggregating science data and generating reports to monitor instrument health, performance, and data quality. Tohban is the Japanese word for "operator" or "duty officer," and in the context of NASA science missions, refers to a scientist whose job is to monitor the science quality of observations and quickly notify the operations team if anything is amiss (sometimes referred to as the Scientist in the Loop (SITL)).

The Autohban aims to automate repetitive work, while still allowing scientists to easily add their own notes. Because of the science collection intents and the DCT, the science processing pipeline can now list all intended science collections (even when the data have yet to be downlinked) and its completeness, putting it all together in context with geomagnetic indicies (AE, Dst, and Kp) to save time for the Tohban.

Dst and Kp indices are easy to pull from the Potsdam and Kyoto servers. However, the daily AE charts provided by Kyoto do not supply the raw data. Instead, our new science processing pipeline pulls the images and uses edge detection to scrape the charts and produce a data product called "Proxy AE". Post-ELFIN, this service has now been moved to the THEMIS mission, and proxy AE can be accessed there.

Weekly (offset by downlink latency), Autohban reports are generated, giving scientists a framework where they only need to mark science zones as one of 5 flags: extremely interesting, potentially interesting, too many gaps (indicating to operators that this science zone needs to be redownlinked), bad data (indicating something wrong with the instruments), and test data (indicating data are used for calibration purposes and are not science worthy). Scientists can also add their own notes (e.g., indicating good conjunctions with other missions, or potential precipitation drivers). These Autohban outputs are evaluated weekly, allowing for a streamlined interface between scientists and operators. Onboard operational issues are therefore quickly found and resolved when they arise, and, compared to before, the operations team can react far more quickly to missing data.

VI. Reflection and Lessons Learned

Ultimately, ELFIN is a small student-driven development and satellite operations team. An oft repeated phrase in spacecraft engineering is to "push complexity to the ground." In this case, however, a simple but flexible flight software design made satellite operations untenable, incurring significant team attrition and resulting in software that was incompatible with the team it was supposed to help. Instead, design should focus on being as simple and understandable as possible, prioritizing these facets at the cost of performance, flexibility, and other factors. Refactor 2.0 was a significant undertaking that vastly simplified the operational paradigm; as the mission neared reentry, the ELFIN ground operations codebase was over 300000 lines of code and comprised of 15 different custom software modules working in tandem. This is a testament to the difficulty of spacecraft operations, which remains deceptively hard even for small spacecraft missions like ELFIN.

Previously, both the science and operations teams barely understood the limitations of the operational software, so communication between the two was ineffective and often counterproductive. Additionally, the only people who understood the software, the developer team, did not understand the needs of the science and operations team. We fixed this by having developers operate and present metrics to the science team even after the initial rollout of Refactor 2.0. This led to a positive feedback cycle that continually fixed operational problems by focusing on metrics, usability, and transparency. It formalized and standardized both operations and scientific oversight, leading to streamlined operations and a significantly larger volume of science data.

Throughout this process, the development team also learned that simplifying the development process should generally be one of the top priorities. A student-based development team has less experience and less motivation to work on software they do not fully understand. For example, layered code abstracts complexity: this is useful when troubleshooting problems because you can easily isolate it from other layers. Layering should occur logically with a clear distinction (e.g., packets $\rightarrow$ science packets $\rightarrow$ EPD packets). At the architectural level, we divided the Planner into three separate layers: the Intent manager (handles all interaction to the database, e.g., obtaining and inserting Intents and Allocations), the constraint checker (determines state and enforces rules and logic on Intents and Allocations), and

the wizard (the GUI for the creation of Intents and Allocations that utilizes the constraint checker). At a more technical level, we followed layering to ensure that database access, for example, never directly used SQL or ORM in the code, relying instead on a layer of helper functions (in a library or several libraries) to act as an interface to the database. This clear delineation made it easier not only for the developer team to review, troubleshoot, and work on the code, but also for the operators to more easily point out areas for improvement.

A final important lesson learned is to account for the evolving nature of scientists' needs (motivated by new findings and the exploratory nature of science missions), which are often in direct conflict with a paradigm focused on simplification. In order to avoid coding ourselves into a corner, we expanded upon the open-closed principle: software entities should be open for extension but closed for modification. This is embodied by our Intents and Allocations paradigm which were easily extended to perform Inner Belt Collections, Ion collections, special 32-sector collections, and perform new attitude maneuvers. For example, the data structure and schema that made up Science Collection Intents were extended multiple times to support each new feature, but always remained backward compatible via versioning within the Planner framework.

ELFIN's successful mission operations was afforded by a large amount of work and dedication from many UCLA students over 4 years with generous support from scientists and limited support from staff. Both flight and ground software for upcoming UCLA missions greatly benefit from the knowledge acquired throughout this refactor endeavor. In order to advance towards the more affordable spacecraft missions of the future, it is crucial to acknowledge that mission operations hold the same level of importance as well-designed instruments and reliable spacecraft engineering. This recognition is essential to achieve meaningful scientific results as the number of small, low-cost science missions are increasing.

VII. ADCS

A benefit to the newly refactored operations is that the team had more time to improve the mission, where possible, by updating and expanding on the preexisting framework. This had led to an expansion of scientific scope as well, including studies of the inner belt and experimentally doubling our EPD's temporal resolution. However, extra effort was also allocated to resolving the two mission level issues: 1) fully calibrating ion data (which occured in June 2022, as ions are considered secondary science) and 2) fixing ELFIN's attitude control.

A. ADCS Overview

ELFIN utilized a highly stripped-down ADCS architecture that relied on only one sensor and a pair of actuators. This is possible due to the spin-stabilization provided by the stacer boom, which damps rotation about other axes other than the primary Z spin axis. Attitude determination is performed on the ground, utilizing data collected from a low-cost, but reliable 3-axis magnetoresisitve magnetometer (MRM). There were two redudntant HMC5883Ls, one placed on

either side of the spacecraft. Spin rate was maintained at 21.15 ± .2 RPM maintained using an air coil (along Y face), using closed-loop b-act and b-dot algorithms. The former relied on applying a torque based on the actual magnetic field, and reliably worked at low spin rates; the latter is widely used for detumbling spacecraft [15]. On ELFIN, b-dot, implemented using FIR filters, was tuned to control spin rates from 7-35 RPM. Spin control worked great due in part to the rigorous testing on the ground before launch.

"Reorientations" (or "reors" for short), on the other hand, could not be tested on the ground and relied on simulation for validation. The spacecraft spin axis could be precessed by pulsing the other air coil (along the Z face) at precise points in orbit, calculated on the ground. A reor would typically require 4 pulses per orbit for $\sim$ 10 orbits to move ELFIN's spin axis about 20°. Reors, however, were unreliable and difficult to verify on orbit due to our uncertainty regarding attitude determination. As operations and obtaining science took the primary development focus initially, ADCS troubleshooting and development efforts were relegated a lower priority. In mid-2021, attitude determination was reinvestgiated for improvements, which eventually led to the successful implementation of attitude control in November 2021.

VIII. Electron Precipitation Studies Enabled by ELFIN

ELFIN has, for the first time, enabled studies of the efficiency (precipitating-to-trapped flux ratio) of different drivers of energetic electron precipitation, from the inner belt all the way to the plasma sheet. Without a doubt, these studies would be impossible without the rigorous testing prior to delivery and heroic team efforts that led to the implementation of Refactor 2.0. Throughout its mission lifetime, ELFIN has accumulated more than 12,500 high-quality radiation belt crossings that are usable for science. This unprecedented coverage of electron precipitation measurements has resulted in studies in essentially every facet of magnetospheric physics that result in electron precipitation. The large volume of data achieved by the revised operations paradigm described herein was critical for many aspects of ELFIN research. For multi-case studies and conjunction studies with equatorial missions, it enabled collections over a wide range of activities, local times and geometrical (spatial) configurations. This variety in observations increases confidence in the results often presented with prototypical examples of the impression gained from examining all available cases. Clearly, statistical studies have also been well served by the availability of a large dataset, enabling ensemble studies as a function of space and activity with statistically significant results. Below, we summarize some of the research enabled by ELFIN in just the last 2.5 years.

Energetic electron precipitation is important to study because it contributes a significant amount of energy input to the atmosphere (especially relativistic electrons penetrating well below 100 km altitude), causing atmospheric changes and is oftentimes correlated with auroral physics (typically driven by the lower-energy range of the precipitating spectrum). Understanding how electrons interact with electromagnetic waves also contributes to the fundamental field of plasma physics. Tsai et al. [16] covers relativistic energetic precipitation by intense field-aligned waves using

magnetic conjunctions between ELFIN and equatorial missions, confirming the validity of ELFIN data. This is followed by statistically analyzing ELFIN-measured precipitation to study the mechanisms by which whistler-mode waves can achieve such energetic precipitation [17, 18]. This study of electorn scattering by whistler-mode waves is complemented by a myriad more studies using ELFIN conjunctions with a variety of spacecraft and ground stations. Artemyev et al. [19] confirmed that ducted whistler-mode waves are likely a significant contributor to relativistic electron losses. An alternative scenario of relativistic precipitation driven by very oblique whistler-mode waves has been considered in Gan et al. [20], where a few events of ELFIN-THEMIS conjunction events in support of this hypothesis were analyzed. These very oblique waves were shown to cause similar (but less intense) relativistic electron precipitation by accounting for higher order resonances. Multiple conjunctions of ELFIN with near-equatorial measurements of very oblique whistler-mode waves also revealed new mechanisms of bursty precipitation of sub-relativistic electrons: nonlinear Landau trapping [21] and the loss-cone overfilling [22].

Storms are the primary dynamic and energetic phenomena in the inner magnetosphere. Operationally, the team would increase the number of collections/day if active times were predicted, and so there are many instances where ELFIN has multiple successive orbits that allow for outer radiation belt scans of the storm evolution at a relatively rapid 90 minute timestep. These consecutive ELFIN observations allow for the estimation of electron life-times and its comparison with state-of-the-art models. Mourenas et al. [23] showed that statistical theoretical lifetime models agree reasonably well with electron pitch-angle diffusion rates inferred from the precipitated-to-trapped 100 keV electron flux ratio measured by ELFIN, as well as with timescales of trapped electron flux decay independently measured over several days by ELFIN. This result demonstrates, for the first time, a broad consistency between timescales of trapped electron flux decay, the pitch-angle distribution of precipitated electrons, and quasi-linear models of wave-driven electron loss, which validates the reliability of such statistical electron lifetime models. Using a similar approach, Mourenas et al. [24] later showed that measured precipitating electron fluxes were well recovered by quasi-linear diffusion models when the wave field consists of short chorus wave packets of moderate amplitudes (160-250 pT), i.e., wave field modulation to sub-packets essentially leads to more diffusive-like transport electrons toward the loss cone than long packets.

Another large source of relativistic precipitation is the interaction of electrons with electromagnetic ion cyclotron (EMIC) waves, which are very efficient in resonating with $>$ 1 MeV electrons and predominately occur at dusk MLT sectors. Angelopoulos et al. [25] comprehensively studied this resonant mechanism using ELFIN to characterize the spatial and temporal distribution of EMIC-driven precipitation and validated theoretical predictions of EMIC wave excitation, dispersion, and damping in multi-ion plasmas. Grach et al. [26] utilized modeling coupled with ELFIN data to highlight the role of nonlinear resonance with EMIC waves, while An et al. [27] demonstrated ELFIN observations of sub-MeV relativistic electron precipitation that could only be explained by non-resonant interactions with EMIC waves. Bashir et al. [28] explored the combination of whistler-mode waves and EMIC waves working in concert in the post-midnight sector, demonstrating the ability of electrons to be first accelerated by whistlers before being

scattered at multi-MEV energies. Finally, Capannolo et al. [29] performed a statistical investigation of ELFIN-observed EMIC-driven precipitation in conjunction with POES observations of EMIC-driven proton precipitations.

Recently, there has been a surge in interest in microbursts: short-lived and intense bursts of electron precipitation that can reach MeV energies but last for typically 100 ms. ELFIN was not originally designed to study microbursts, but we have shown that ELFIN can resolve them [30] when ELFIN observes intense precipitation at the sub-spin time resolution (in fact, within a single sector). For a long time, it was unclear whether microbursts were caused by EMIC and whistler-mode waves or whistler-mode waves alone. However, Chen et al. [31] used modeling and a conjunction with the Van Allen probes to clearly show that the most probable mechanism for ELFIN-observed microburst generation is electron resonance with ducted whistler-mode waves. The geographic occurrence of microbursts, along with their spatial and temporal scales, was further constrained by Zhang et al. [32], and exhibited many similarities with the origins of whistler-mode waves.

Despite being less dynamic, field line curvature (FLC) scattering is also a major loss mechanism (and therefore atmospheric energy source) and occurs when the field curvature radius becomes comparable to the particle gyroradius, resulting in a violation of the first adiabatic invariant and subsequent pitch-angle scattering. When the magnetic fields are no longer able to constrain particles, the entire energy range above a specific minimum energy becomes isotropic at a particular latitude. The dependence of such minimum resonant energy on latitude (i.e. on both equatorial magnetic field intensity and magnetic field line curvature radius) yields an energy dispersion seen in FLC scattering patterns. That energy-latitude dependence marks the "isotropy boundary" (IB), of which their precipitation contribution had never been measured with decent energy resolution prior to ELFIN. Wilkins et al. [33] used the entire ELFIN data set of electron loss measurements to remotely characterize the properties and locations of IBs and found that they were responsible for 10%-20% (10 MW) of the total average electron precipitation power. In fact, IB energy deposition during active times could reach nearly 100% (1 GW) of the total electron energy deposition across the entire subauroral and auroral zone.

Any mechanism that can cause electron precipitation can be remotely studied using ELFIN. Shen et al. [34, 35] investigated electron losses driven by kinetic Alfven waves during substorms and plasma injections, while Shi et al. [36] demonstrated how ULF modulates whistler-mode waves, causing quasi-periodic electron precipitation. Because of the Refactor, ELFIN was able to expand its data acquisition from occasional to routine measurements of the inner belt as well, enabling studies of particle lifetimes and phenomena at lower L-shells. Although the inner belt is more stable, Shen et al. [37] found many instances of characteristic WISP precipitation caused by interactions with VLF transmitters. Because these short radiation belt crossings have equatorial footprints that span from the plasmasphere all the way to the plasmasheet, ELFIN's observations provide a radial snapshot of the entire magnetosphere, which is a uniquely rich data set that is applicable to almost every mangetospheric process, mapping anywhere from the inner radiation belt to the polar cap.

IX. Conclusion

Despite ELFIN's four successful years on orbit, there were many points throughout the last decade when ELFIN's success seemed entirely out of reach. After successful post-launch operations (acquisition, spin up, and commissioning), a difficult battle of attrition shortly followed where students rotated and burned out too frequently, and motivation was hard to come by as mission operations became tedious, error-prone, and unrewarding. Refactor 2.0 marked a radical rethinking of satellite operations and significantly boosted downlinked science volume while reducing operational workload. This, in turn, provided the necessary statistics to confidently scale our science processing data pipeline. Improved coordination between software developers, spacecraft operators, and the science team enabled an expandable architecture that led to the sustainable relationship between mission operators and scientists. It still took another full year of software development, data analysis, and on-orbit instrument calibration, before ELFIN data was deemed science worthy. There are now several studies that show the validity and quality of ELFIN data sets using equatorial comparisons [16, 17, 25, 38]. As of this writing, there have been over 40 ELFIN-enabled publications in scientific peer-reviewed journals, making ELFIN one of the most scientifically productive CubeSat missions across both NASA and NSF's entire CubeSat fleet to date. ELFIN is the strongest example yet of low-cost, high-impact science, made possible by a passionate and dedicated team of students, and enabled by the mentorship of UCLA EPSS staff and partnerships with industry.

Open Research

ELFIN data is available at `https://data.elfin.ucla.edu/` and online summary plots are available at `https://plots.elfin.ucla.edu/summary.php`.

Acknowledgments and Funding Sources

We are grateful to NASA's CubeSat Launch Initiative and Launch Services Program for ELFIN's successful launch in the desired orbits. We acknowledge early support of ELFIN project by the AFOSR, under its University Nanosat Program, UNP-8 project, contract FA9453-12-D-0285, and by the California Space Grant program. We acknowledge support from NASA grant NNX14AN68G and NSF grants AGS-1242918 and AGS-2019950 which provided vital funding for the development and early operations of ELFIN. Importantly, we acknowledge the critical contributions by hundreds of UCLA students who made the ELFIN mission a success.

References

[1] Horne, R. B., Thorne, R. M., Glauert, S. A., Meredith, N. P., Pokhotelov, D., and Santolík, O., "Electron acceleration in the Van Allen radiation belts by fast magnetosonic waves," , Vol. 34, 2007, p. 17107. https://doi.org/10.1029/2007GL030267.

[2] Horne, R. B., Glauert, S. A., Meredith, N. P., Boscher, D., Maget, V., Heynderickx, D., and Pitchford, D., "Space weather

impacts on satellites and forecasting the Earth's electron radiation belts with SPACECAST," *Space Weather*, Vol. 11, 2013, pp. 169–186. https://doi.org/10.1002/swe.20023.

[3] Baker, D. N., Erickson, P. J., Fennell, J. F., Foster, J. C., Jaynes, A. N., and Verronen, P. T., "Space Weather Effects in the Earth's Radiation Belts," *Space Science Reviews*, Vol. 214, No. 1, 2017, p. 17. https://doi.org/10.1007/s11214-017-0452-7, URL https://doi.org/10.1007/s11214-017-0452-7.

[4] Miyoshi, Y., Saito, S., Kurita, S., Asamura, K., Hosokawa, K., Sakanoi, T., Mitani, T., Ogawa, Y., Oyama, S., Tsuchiya, F., Jones, S. L., Jaynes, A. N., and Blake, J. B., "Relativistic Electron Microbursts as High-Energy Tail of Pulsating Aurora Electrons," , Vol. 47, No. 21, 2020, e90360. https://doi.org/10.1029/2020GL090360.

[5] Miyoshi, Y., Hosokawa, S., Kurita, S.-I., Oyama, Y., Ogawa, S., Saito, I., Shinohara, A., Kero, E., Turunen, P. T., Verronen, S., Kasahara, S., Yokota, T., Mitani, T., Takashima, N., Higashio, Y., Kasahara, S., Masuda, F., Tsuchiya, A., Kumamoto, A., Matsuoka, T., Hori, K., Keika, M., Shoji, M., Teramoto, S., Imajo, C., Jun, S., and Nakamura, "Penetration of MeV electrons into the mesosphere accompanying pulsating aurorae," *Scientific Reports*, Vol. 11, 2021, p. 13724. https://doi.org/10.1038/s41598-021-92611-3.

[6] Angelopoulos, V., Tsai, E., Bingley, L., Shaffer, C., Turner, D. L., Runov, A., Li, W., Liu, J., Artemyev, A. V., Zhang, X. J., Strangeway, R. J., Wirz, R. E., Shprits, Y. Y., Sergeev, V. A., Caron, R. P., Chung, M., Cruce, P., Greer, W., Grimes, E., Hector, K., Lawson, M. J., Leneman, D., Masongsong, E. V., Russell, C. L., Wilkins, C., Hinkley, D., Blake, J. B., Adair, N., Allen, M., Anderson, M., Arreola-Zamora, M., Artinger, J., Asher, J., Branchevsky, D., Capitelli, M. R., Castro, R., Chao, G., Chung, N., Cliffe, M., Colton, K., Costello, C., Depe, D., Domae, B. W., Eldin, S., Fitzgibbon, L., Flemming, A., Fox, I., Frederick, D. M., Gilbert, A., Gildemeister, A., Gonzalez, A., Hesford, B., Jha, S., Kang, N., King, J., Krieger, R., Lian, K., Mao, J., McKinney, E., Miller, J. P., Norris, A., Nuesca, M., Palla, A., Park, E. S. Y., Pedersen, C. E., Qu, Z., Rozario, R., Rye, E., Seaton, R., Subramanian, A., Sundin, S. R., Tan, A., Turner, W., Villegas, A. J., Wasden, M., Wing, G., Wong, C., Xie, E., Yamamoto, S., Yap, R., Zarifian, A., and Zhang, G. Y., "The ELFIN Mission," , Vol. 216, No. 5, 2020, 103. https://doi.org/10.1007/s11214-020-00721-7.

[7] Angelopoulos, V., "The THEMIS Mission," , Vol. 141, 2008, pp. 5–34. https://doi.org/10.1007/s11214-008-9336-1.

[8] Burch, J. L., Moore, T. E., Torbert, R. B., and Giles, B. L., "Magnetospheric Multiscale Overview and Science Objectives," , Vol. 199, 2016, pp. 5–21. https://doi.org/10.1007/s11214-015-0164-9.

[9] Mauk, B. H., Fox, N. J., Kanekal, S. G., Kessel, R. L., Sibeck, D. G., and Ukhorskiy, A., "Science Objectives and Rationale for the Radiation Belt Storm Probes Mission," , Vol. 179, 2013, pp. 3–27. https://doi.org/10.1007/s11214-012-9908-y.

[10] Kasahara, S., Miyoshi, Y., Yokota, S., Kasahara, Y., Matsuda, S., Kumamoto, A., Matsuoka, A., Kazama, Y., Frey, H. U., Angelopoulos, V., Kurita, S., Keika, K., Seki, K., and Shinohara, I., "Pulsating aurora from electron scattering by chorus waves," *Nature*, Vol. 554, 2018, pp. 337–340. https://doi.org/10.1038/nature25505.

[11] Baker, D., Mason, G., Figueroa, O., Colon, G., Watzin, J., and Aleman, R., "An overview of the Solar Anomalous, and Magnetospheric Particle Explorer (SAMPEX) mission," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 31, No. 3, 1993, pp. 531–541. https://doi.org/10.1109/36.225519.

[12] Evans, D. S., and Greer, M. S., "Polar orbiting environmental satellite space environment monitor-2: instrument description and archive data documentation," , 2004.

[13] Crew, A. B., Spence, H. E., Blake, J. B., Klumpar, D. M., Larsen, B. A., O'Brien, T. P., Driscoll, S., Handley, M., Legere, J., Longworth, S., Mashburn, K., Mosleh, E., Ryhajlo, N., Smith, S., Springer, L., and Widholm, M., "First multipoint in situ observations of electron microbursts: Initial results from the NSF FIREBIRD II mission," *Journal of Geophysical Research (Space Physics)*, Vol. 121, No. 6, 2016, pp. 5272–5283. https://doi.org/10.1002/2016JA022485.

[14] Aumasson, J.-P., and Bernstein, D. J., "SipHash: A Fast Short-Input PRF," 2012, pp. 489–508.

[15] Lovera, M., "Magnetic satellite detumbling: The b-dot algorithm revisited," *2015 American Control Conference (ACC)*, 2015, pp. 1867–1872. https://doi.org/10.1109/ACC.2015.7171005.

[16] Tsai, E., Artemyev, A., Zhang, X.-J., and Angelopoulos, V., "Relativistic Electron Precipitation Driven by Nonlinear Resonance With Whistler-Mode Waves," *Journal of Geophysical Research (Space Physics)*, Vol. 127, No. 5, 2022, e30338. https://doi.org/10.1029/2022JA030338.

[17] Tsai, E., Artemyev, A., Angelopoulos, V., and Zhang, X.-J., "Investigating Whistler-Mode Wave Intensity Along Field Lines Using Electron Precipitation Measurements," *Journal of Geophysical Research (Space Physics)*, Vol. 128, No. 8, 2023, e2023JA031578. https://doi.org/10.1029/2023JA031578.

[18] Tsai, E., Artemyev, A. V., Ma, Q., Mourenas, D., Agapitov, O., Zhang, X.-J., and Angelopoulos, V., "Key factors determining nightside energetic electron losses driven by whistler-mode waves," 2023. https://doi.org/10.22541/au.170216582.29158404/v1.

[19] Artemyev, A. V., Demekhov, A. G., Zhang, X. J., Angelopoulos, V., Mourenas, D., Fedorenko, Y. V., Maninnen, J., Tsai, E., Wilkins, C., Kasahara, S., Miyoshi, Y., Matsuoka, A., Kasahara, Y., Mitani, T., Yokota, S., Keika, K., Hori, T., Matsuda, S., Nakamura, S., Kitahara, M., Takashima, T., and Shinohara, I., "Role of Ducting in Relativistic Electron Loss by Whistler-Mode Wave Scattering," *Journal of Geophysical Research (Space Physics)*, Vol. 126, No. 11, 2021, e29851. https://doi.org/10.1029/2021JA029851.

[20] Gan, L., Li, W., Ma, Q., Artemyev, A. V., and Albert, J. M., "Dependence of Nonlinear Effects on Whistler-Mode Wave Bandwidth and Amplitude: A Perspective From Diffusion Coefficients," *Journal of Geophysical Research (Space Physics)*, Vol. 127, No. 5, 2022, e30063. https://doi.org/10.1029/2021JA030063.

[21] Artemyev, A. V., Zhang, X. J., Zou, Y., Mourenas, D., Angelopoulos, V., Vainchtein, D., Tsai, E., and Wilkins, C., "On the Nature of Intense Sub-Relativistic Electron Precipitation," *Journal of Geophysical Research (Space Physics)*, Vol. 127, No. 6, 2022, e30571. https://doi.org/10.1029/2022JA030571.

[22] Zhang, X.-J., Artemyev, A., Angelopoulos, V., Tsai, E., Wilkins, C., Kasahara, S., Mourenas, D., Yokota, S., Keika, K., Hori, T., Miyoshi, Y., Shinohara, I., and Matsuoka, A., "Superfast precipitation of energetic electrons in the radiation belts of the Earth," *Nature Communications*, Vol. 13, 2022, 1611. https://doi.org/10.1038/s41467-022-29291-8.

[23] Mourenas, D., Artemyev, A. V., Zhang, X. J., Angelopoulos, V., Tsai, E., and Wilkins, C., "Electron Lifetimes and Diffusion Rates Inferred From ELFIN Measurements at Low Altitude: First Results," *Journal of Geophysical Research (Space Physics)*, Vol. 126, No. 11, 2021, e29757. https://doi.org/10.1029/2021JA029757.

[24] Mourenas, D., Zhang, X. J., Nunn, D., Artemyev, A. V., Angelopoulos, V., Tsai, E., and Wilkins, C., "Short Chorus Wave Packets: Generation Within Chorus Elements, Statistics, and Consequences on Energetic Electron Precipitation," *Journal of Geophysical Research (Space Physics)*, Vol. 127, No. 5, 2022, e30310. https://doi.org/10.1029/2022JA030310.

[25] Angelopoulos, V., Zhang, X.-J., Artemyev, A. V., Mourenas, D., Tsai, E., Wilkins, C., Runov, A., Liu, J., Turner, D. L., Li, W., Khurana, K., Wirz, R. E., Sergeev, V. A., Meng, X., Wu, J., Hartinger, M. D., Raita, T., Shen, Y., An, X., Shi, X., Bashir, M. F., Shen, X., Gan, L., Qin, M., Capannolo, L., Ma, Q., Russell, C. L., Masongsong, E. V., Caron, R., He, I., Iglesias, L., Jha, S., King, J., Kumar, S., Le, K., Mao, J., McDermott, A., Nguyen, K., Norris, A., Palla, A., Roosnovo, Tam, J., Xie, E., Yap, R. C., Ye, S., Young, C., Adair, L. A., Shaffer, C., Chung, M., Cruce, P., Lawson, M., Leneman, D., Allen, M., Anderson, M., Arreola-Zamora, M., Artinger, J., Asher, J., Branchevsky, D., Cliffe, M., Colton, K., Costello, C., Depe, D., Domae, B. W., Eldin, S., Fitzgibbon, L., Flemming, A., Frederick, D. M., Gilbert, A., Hesford, B., Krieger, R., Lian, K., McKinney, E., Miller, J. P., Pedersen, C., Qu, Z., Rozario, R., Rubly, M., Seaton, R., Subramanian, A., Sundin, S. R., Tan, A., Thomlinson, D., Turner, W., Wing, G., Wong, C., and Zarifian, A., "Energetic electron precipitation driven by electromagnetic ion cyclotron waves from ELFIN's low altitude perspective," *Space Science Reviews*, Vol. 219, No. 37, 2023. https://doi.org/10.1007/s11214-023-00984-w.

[26] Grach, V. S., Artemyev, A. V., Demekhov, A. G., Zhang, X.-J., Bortnik, J., Angelopoulos, V., Nakamura, R., Tsai, E., Wilkins, C., and Roberts, O. W., "Relativistic Electron Precipitation by EMIC Waves: Importance of Nonlinear Resonant Effects," , Vol. 49, No. 17, 2022, e99994. https://doi.org/10.1029/2022GL099994.

[27] An, X., Artemyev, A., Angelopoulos, V., Zhang, X., Mourenas, D., and Bortnik, J., "Nonresonant Scattering of Relativistic Electrons by Electromagnetic Ion Cyclotron Waves in Earth's Radiation Belts," , Vol. 129, No. 13, 2022, 135101. https://doi.org/10.1103/PhysRevLett.129.135101.

[28] Bashir, M. F., Artemyev, A. V., Zhang, X.-J., Angelopoulos, V., Tsai, E., and Wilkins, C., "Observations of relativistic electron precipitation due to combined scattering of whistler-mode and EMIC waves," *Authorea Preprints*, 2023.

[29] Capannolo, L., Li, W., Ma, Q., Qin, M., Shen, X. C., Angelopoulos, V., Artemyev, A., Zhang, X. J., and Hanzelka, M., "Electron Precipitation Observed by ELFIN Using Proton Precipitation as a Proxy for Electromagnetic Ion Cyclotron (EMIC) Waves," , Vol. 50, No. 21, 2023, e2023GL103519. https://doi.org/10.1029/2023GL103519.

[30] Zhang, X.-J., Angelopoulos, V., Mourenas, D., Artemyev, A., Tsai, E., and Wilkins, C., "Characteristics of Electron Microburst Precipitation Based on High-Resolution ELFIN Measurements," *Journal of Geophysical Research (Space Physics)*, Vol. 127, No. 5, 2022, e30509. https://doi.org/10.1029/2022JA030509.

[31] Chen, L., Zhang, X.-J., Artemyev, A., Angelopoulos, V., Tsai, E., Wilkins, C., and Horne, R. B., "Ducted Chorus Waves Cause Sub-Relativistic and Relativistic Electron Microbursts," , Vol. 49, No. 5, 2022, e97559. https://doi.org/10.1029/2021GL097559.

[32] Zhang, X.-J., Angelopoulos, V., Artemyev, A., Mourenas, D., Agapitov, O., Tsai, E., and Wilkins, C., "Temporal Scales of Electron Precipitation Driven by Whistler-Mode Waves," *Journal of Geophysical Research (Space Physics)*, Vol. 128, No. 1, 2023, e2022JA031087. https://doi.org/10.1029/2022JA031087.

[33] Wilkins, C., Angelopoulos, V., Runov, A., Artemyev, A., Zhang, X. J., Liu, J., and Tsai, E., "Statistical Characteristics of the Electron Isotropy Boundary," *Journal of Geophysical Research (Space Physics)*, Vol. 128, No. 10, 2023, e2023JA031774. https://doi.org/10.1029/2023JA031774.

[34] Shen, Y., Artemyev, A. V., Zhang, X.-J., Angelopoulos, V., Vasko, I., Turner, D., Tsai, E., Wilkins, C., Weygand, J. M., Russell, C. T., Ergun, R. E., and Giles, B. L., "Tens to Hundreds of keV Electron Precipitation Driven by Kinetic Alfvén Waves During an Electron Injection," *Journal of Geophysical Research (Space Physics)*, Vol. 127, No. 8, 2022, e30360. https://doi.org/10.1029/2022JA030360.

[35] Shen, Y., Artemyev, A. V., Zhang, X.-J., Zou, Y., Angelopoulos, V., Vasko, I., Runov, A., Tsai, E., and Wilkins, C., "Contribution of Kinetic Alfvén Waves to Energetic Electron Precipitation From the Plasma Sheet During a Substorm," *Journal of Geophysical Research (Space Physics)*, Vol. 128, No. 4, 2023, e2023JA031350. https://doi.org/10.1029/2023JA031350.

[36] Shi, X., Zhang, X.-J., Artemyev, A., Angelopoulos, V., Hartinger, M. D., Tsai, E., and Wilkins, C., "On the Role of ULF Waves in the Spatial and Temporal Periodicity of Energetic Electron Precipitation," *Journal of Geophysical Research (Space Physics)*, Vol. 127, No. 12, 2022, e2022JA030932. https://doi.org/10.1029/2022JA030932.

[37] Shen, Y., Artemyev, A. V., Ma, Q., Zhang, X.-J., Mourenas, D., Tsai, E., Wilkins, C., Wu, J., and Angelopoulos, V., "Inner Belt Wisp Precipitation Measured by ELFIN: Regimes of Energetic Electron Scattering by VLF Transmitter Waves," *Journal of Geophysical Research (Space Physics)*, Vol. 127, No. 11, 2022, e2022JA030968. https://doi.org/10.1029/2022JA030968.

[38] Artemyev, A. V., Angelopoulos, V., Zhang, X. J., Runov, A., Petrukovich, A., Nakamura, R., Tsai, E., and Wilkins, C., "Thinning of the Magnetotail Current Sheet Inferred From Low-Altitude Observations of Energetic Electrons," *Journal of Geophysical Research (Space Physics)*, Vol. 127, No. 10, 2022, e2022JA030705. https://doi.org/10.1029/2022JA030705.