Enabling Resilient and Autonomous Collection of Near-Earth Objects

Cameron Harris, Sean Crosby, Bryce Castle,
Vishvarath Balasubramanian-Karthikeyan, Stella Yang
Sandia National Laboratories

ABSTRACT

The increased use of space for commercial and military applications presents a challenge in maintaining accurate and timely Space Situational Awareness (SSA). Sandia National Laboratories is actively developing and enhancing a fully autonomous pipeline for SSA data acquisition and exploitation. The system, known as the SSA Observatory, comprises of a telescope mount, camera, and observatory dome. This effort's goal is to facilitate extensive SSA data collection and provide rich metadata for future data exploitation efforts.

The SSA Observatory enables rapid SSA data acquisition and exploitation with minimal human-in-the-loop interaction. SSA Observatory autonomously schedules and collects images of satellites based on visibility and assigned priority. The system supports two modes of data collection: direct tasking by end-users and uncued collection of known visible targets. To enable unmanned collections, the system is designed for resilience against hazardous conditions. A weather sensor provides real-time safety monitoring, and actively checks local weather, including precipitation, wind, and temperature. An accompanying monitoring system ensures that critical processes run and warns when conditions are unsound. Provided in this paper is an analysis of design decisions, discussion of challenges, and presentation of sky imagery.

1. INTRODUCTION

The growing number of catalogued objects in the space domain requires sufficient sensing resources to maintain comprehensive awareness. A recent report by the United States Government Accountability Office explains the nation's need for improved space surveillance [16]. The report posits that the advancement of SSA will require considerable effort from both government and industry, and data is a key component of preserving and expanding SSA. The development of advanced technologies may require real sky imagery data for training, testing, or validation of algorithms. However, existing data can be proprietary, restricted, bespoke, or some combination of the three. Moreover, sky imagery is difficult to obtain extensively and consistently, in part due to unamenable weather, hardware upkeep, operator needs, and physical limits on observation opportunities.

Sandia National Laboratories stood up the SSA Observatory to address challenges related to SSA data acquisition and exploitation. Presently, the SSA Observatory seeks to capture and produce as much nightly SSA data as possible, which is enabled by system-wide automation. Limited manpower demands that the system operates safely and reliably without human intervention – the goal is that an operator should not be required to maintain critical processes, protect hardware, or otherwise provide assistance during nightly collections.

Automation is not a new concept in SSA data collection. Many SSA data collection systems implement some level of automation; a quintessential example is the FireOPAL observatories in Australia [2]. FireOPAL observatories are standalone, fully automated optical sensors used to track passing satellites in near-Earth orbit. The systems are weather-hardened, and do not require a protective enclosure like a dome.

The intent of this work is to walk through Sandia's design choices, challenges, and solutions to achieve full automation, with the expectation that principles applied and lessons learned will provide value to the larger community. In the SSA Observatory's pursuit of unmanned SSA data collection, four overarching requirements have been identified:

1. Autonomous hardware operation
2. Automated/asynchronous task scheduling

3. Systematic safe collection conditions
4. Automated failure management and resiliency strategies

The SSA Observatory has developed a sophisticated system, encompassing astronomy hardware, control software, and monitoring policies, to address these four requirements for unmanned SSA data collection. The work presented in this paper will expand on the system developments that address the requirements. The paper proceeds as follows: Section 2 overviews the hardware components and software systems used to facilitate automation. Section 3 outlines the procedure for automatically and/or asynchronously scheduling tasks for SSA data collection. Section 4 describes the safety monitoring infrastructure, principles for identifying hazardous weather conditions, and resiliency strategies. Section 5 presents and analyzes data collected with the SSA Observatory system.

2. HARDWARE CONTROL AND AUTOMATION

2.1 System hardware

The system hardware is comprised of:

- Astro Haven 12' clamshell dome
- Boltwood Cloud Sensor II weather station
- NFOV telescope (Sohbrit)
  - Astro Physics 1600 GTO telescope mount
  - telescope tube
    - 80mm telescope
    - 11in telescope
    - 16in telescope
  - ZWO ASI 1600MM-Cool camera
  - FocusLynx focuser
  - Starlight Xpress filter wheel
- WFOV Sensor (WASSAT)
  - COTS 4 x 1 EO Camera array

All hardware is Commercial-Off-The-Shelf. The telescope, comprised of the mount, camera, focuser, filter wheel, and tube, is known as Sohbrit. The array of static WFOV cameras is WASSAT, which stares at the GEO belt. Sohbrit and WASSAT both sit inside the Astro Haven 12' dome. A picture of Sohbrit inside the dome is provided in Figure 1. WASSAT is too short to be visible in the picture.

Additionally, two computers operate the hardware in the dome. One computer controls Sohbrit, the Boltwood Cloud Sensor II, and the Astro Haven dome. The other computer controls WASSAT.

2.2 Software architecture

Sohbrit contributed to several missions in the past, such as spectral fingerprinting in Reference [3]. While the present automation capabilities are conceptually similar to previous efforts, the underlying software architecture has undergone significant changes and improvements. Specifically, due to challenges with system stability, the automation software has been entirely rewritten and re-envisioned in Python.

Sohbrit, WASSAT, and the dome are each controlled by services that exchange information via HTTP requests. A fourth service, known as the Choreographer, functions as the tasker/scheduler, and distributes tasks to Sohbrit and WASSAT at execution time. The entirety of the software architecture is written in Python; each service is its own Flask server [8]. The services run on a Windows platform.

Fig. 1: SSA Observatory, 2016

Astronomy Common Object Model (ASCOM) drivers are used to control the hardware, which are Windows COM objects. The ASCOM drivers provide a comprehensive and well-documented API specification. The pywin32 package enables Python software to interface with the ASCOM drivers [9]. The Astro Physics 1600 GTO telescope mount, ASI 1600MM camera, FocusLynx focuser, and Starlight Xpress filter wheel all support ASCOM interfaces.

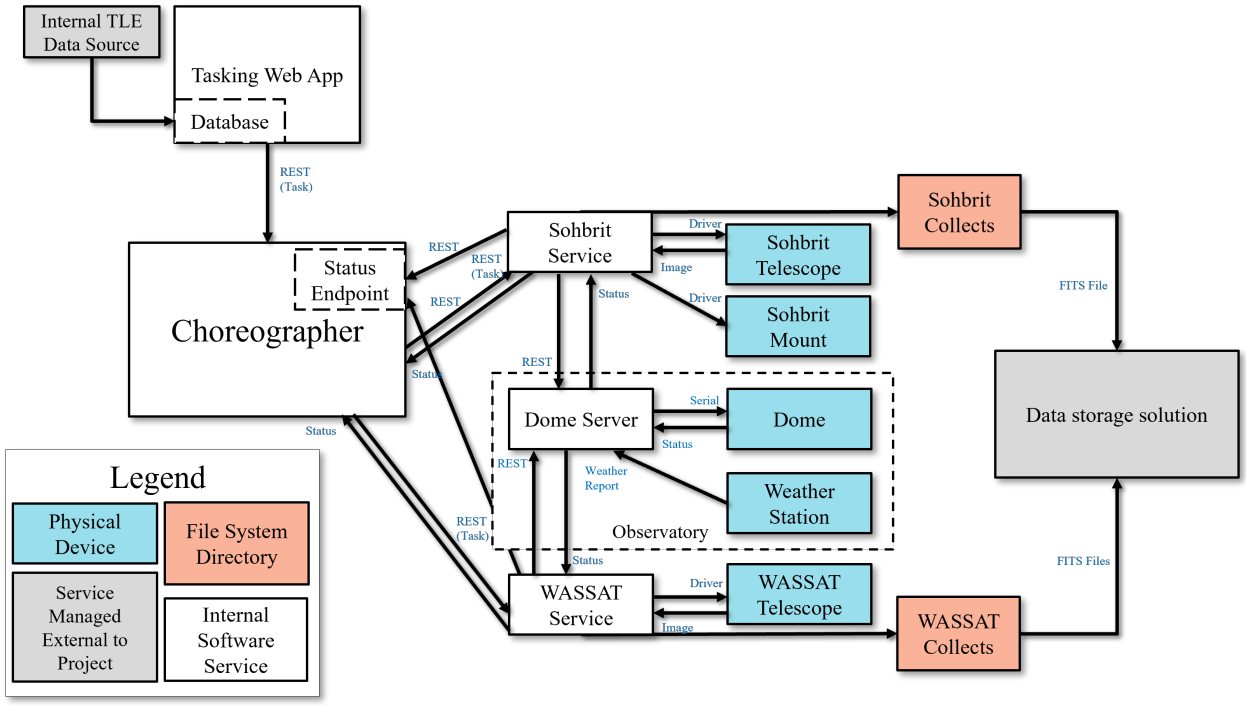


Fig. 2: SSA Observatory software architecture

The Choreographer makes an HTTP POST request to the Sohbrit or WASSAT service containing the task data. In turn, Sohbrit and WASSAT POST request the dome service to open the dome for the expected duration of the task. The dome service may reject the request if the conditions are unsafe for imaging – Section 4 provides information on conditions monitoring. If the dome open request is accepted, the collector proceeds with the collection. Otherwise, the task is considered a failure and no action is taken. Sohbrit and WASSAT report the success of the task back to the

Choreographer, and a new task will be queued if one is available. The software architecture is illustrated in Figure 2.

3. TASK SCHEDULING

Sohbrit supports collection for all near-Earth orbital regimes, including low-earth orbit (LEO). Captured images are saved in FITS format, and image files include metadata in the FITS headers to facilitate downstream exploitation. Sohbrit images in the visible spectrum, and the system can down-select spectral bands using an attached filter wheel. Imagery metadata includes mount pointing angles, capture epoch, integration time, and instantaneous weather data, which are needed for orbit determination and object characterization.

3.1 Target visibility

The SSA Observatory collects on a target during a window of time where the target is expected to be visible. A target's visibility is determined from a sequence of checks. This subsection outlines the procedure for determining object visibility.

The first condition for object visibility is a simple geometry check. Specifically, the object's position must exceed a minimum elevation, el_{min}, from the horizon in order to be viewable. Following the guidance of Reference [17], the minimum elevation threshold is $el_{min} = 15^{\circ}$. A 15° threshold ensures that objects along the horizon will not obstruct line the of sight and that atmospheric effects do not significantly distort the object's apparent position.

Objects eclipsed by the Earth will not be detectable to an optical sensor, so a necessary condition for visibility is that sunlight must illuminate the target. A simple method to filter out eclipsed targets is to first check if a line of sight exists between the target and the Sun. The LIGHT algorithm in Reference [17] demonstrates this calculation. If the target is illuminated, then other conditions can be checked to determine if the target is detectable.

If the target's effective diameter d is known, its apparent brightness can be estimated. The procedure to compute apparent brightness is adapted from the calculations outlined in Reference [10], and the salient steps are shown here. First, the object's total reflectance, $f_{total}(\phi)$, is sum of its diffuse and specular reflectance; i.e.,

$$f_{total}(\phi) = f_{specular}(\phi) + f_{diffuse}(\phi) \tag{1}$$

where ϕ is the Sun-Target-Observer Angle (STOA).

The diffuse reflectance is computed as shown in Equation 2.

$$f_{diffuse}(\phi) = \frac{2}{3\pi^2}(sin(\phi) + (\pi - \phi)cos(\phi)) \tag{2}$$

The specular reflectance is treated as constant:

$$f_{specular}(\phi) = f_{specular} = \frac{1}{4\pi} \tag{3}$$

The Optical Cross Section (OCS) of the object is a proxy for the amount of light reflected by an object that reaches an observer. Equation 4 defines the OCS as a function of the object's physical properties, STOA, and range.

$$OCS = \frac{\pi d^2 \alpha_g f_{total}(\phi)}{4R^2} \tag{4}$$

where α_g is the geometric albedo of the target, d is the effective diameter of the target, and R is the range from the observer to the target. Reference [12] suggests a geometric albedo value of $\alpha_g = 0.175$ is a suitable assumption for space objects.

Finally, the apparent brightness m_{app} is given by Equation 5.

$$m_{app} = M_{\odot} - 2.5log_{10}(OCS) \tag{5}$$

Table 1: Task Structure

Field	Description
Task ID	Unique ID associated with the task
Task creation time	ISO8601-formatted timestamp of the file creation time
NORAD ID	NORAD satellite catalog ID of the desired target
NORAD name	Catalogued name of the desired target
TLE	Two Line Element set associated with the desired target
Rise time	UTC ISO8601-formatted timestamp of the time that the target enters view
Set time	UTC ISO8601-formatted timestamp of the time that the target exits view
Priority	Numerical priority level of the task
Collector options	Specific settings for the selected collector

where $M_{\odot} = -26.74$ is the magnitude of the Sun.

Ultimately, the detectability of the target is determined by comparing the apparent brightness m_{app} against the limiting magnitude of the system; i.e., the target is visible if $m_{app} \leq m_{limiting}$. Sohbrit's $80mm$ wide field of view telescope has been found to reliably image targets with a limiting magnitude setting of $m_{limiting} = 8$. While Sohbrit is able to image objects dimmer than an apparent brightness of 8, imperfect reflectance models and an in-progress streak detection algorithm provide incentive for a bright limiting magnitude setting for the present. Even at a threshold of 8, more than a hundred observation opportunities are consistently available each night. Further discussion of nightly collection and streak detection is provided in Section 5.

If the target's effective diameter is not known, a simple check of the STOA against a configured threshold may suffice. Reference [4] suggests an appropriate limit for the STOA for objects in geosynchronous Earth orbit (GEO) is $\phi \leq 85^{\circ}$. Without noticeable loss in efficacy, 85° limit can be extended to all catalogued objects.

3.2 Task framework

If an object is predicted to be visible, a task may be submitted to the SSA Observatory to observe it. A task is a JSON-format data file which contains the necessary information to track a target. The fields of a task are described in Table 1.

Tasks may be submitted for both Sohbrit and WASSAT. WASSAT is stationary and does not actively track targets, but it can image targets as they pass through the field of view. Sohbrit actively tracks targets, following an open-loop tracking scheme. The target position is propagated to the current time using the ephemeris contained in the task data. Sohbrit will slew to the propagated position and image the target until the batch size is met or the set time passes. By default, each task requests that a batch of ten images are collected of the target.

Target ephemeris are encoded using TLEs. TLEs can be queried from Space-Track (www.space-track.org) with an account. A target's TLE is propagated during collection using the Skyfield Python package [13]. A buffer of 10 seconds is added between image exposures, allowing Sohbrit ample time to propagate the TLE and slew to the next capture point without losing custody of the target.

The SSA Observatory uses an enumerated priority scheme for tasking, where 0 is the lowest priority and 9 is the highest priority. If tasks are of equal priority level, the target with the earlier set time will take precedence. Ranking by set time is useful because it ensures that targets with short flyovers, like LEO objects, are imaged before targets with longer passes.

3.3 Task generation

The SSA Observatory supports two modalities of task scheduling: User requests and autonomous generation. Both modalities are used to generate task files. The task file is deposited in the Choreographer inbox and remains there until execution time. Because TLEs are updated regularly, the rise and set times of the tasks in the Choreographer inbox are adjusted automatically as new TLEs become available.

The process for direct tasking by users is outlined in Subsection 3.3.1; autonomous task generation is outlined in Subsection 3.3.2.

3.3.1 User task requests

A web app accompanies the system where users can submit a task request for a desired target. When a target is requested, the web app presents a list time windows where the target is expected to be visible, based on the conditions described in Subsection 3.1. The user must select a visible window for their task to be executed. While any target in the NORAD catalog is valid for a task request, not all targets are detectable due to the physical limitations of the sensors. If no visible windows are expected for an object, then the object cannot be tasked. Consequently, small pieces of debris and dim objects are currently unavailable for tasking.

For objects that will be visible, all visible windows for the next week listed for selection. Each window is coupled with weather forecast data from National Weather Service to inform the user of the expected weather activity during the window. An unfavorable weather forecast will not prevent a task from being submitted for that window, but unsafe weather conditions at the time of the tasking window will prevent the task from being executed. Section 4 provides additional information on weather management during tasks.

Users may specify other parameters of the task, such as exposure time, gain, and batch size. Because the ZWO ASI-1600MM camera has low read noise, gain defaults to zero if not specified. The user has the ability to set an exposure time for each image, but it is not required. If the user opts not to choose the exposure time, it is computed automatically from the geometry of the flyover. Subsection 3.4 describes the procedure for calculating a sufficient exposure time from flyover geometry.

3.3.2 Autonomous task generation

Nightly tasks can be generated from the entire NORAD catalog, based on the procedure outlined in Subsection 3.1. The procedure is codified in a command line tool, which can be called by Windows Task Scheduler daily. The procedure is simply run on a loop over every target in the catalog for the following night, and a task is generated for every visible window.

Autonomously generated tasks have constant gain and batch size settings; the gain is set to zero (again due to low camera read noise) and the batch size is set to ten. It is prudent to attempt ten images per target because not all images in the batch will be resolvable. A batch of ten images greatly improves the odds that there is sufficient data to perform orbit determination or other characterization analyses. Autonomously generated tasks are assigned the lowest priority level to ensure that user requested tasks take precedence. The exposure time for each autonomous task is computed following the procedure in Subsection 3.4.

3.4 Systematic exposure time

The predicted geometry of the target flyover can be leveraged to systematically determine a suitable exposure time for each image during a collection. Flyover geometry depends on the relative state of the target from the observing sensor, $\vec{x}_{rel}$. The relative state is the combined relative position and velocity, such that:

$$\vec{x}_{rel} = \begin{bmatrix} \vec{r}_{rel} \\ \dot{\vec{r}}_{rel} \end{bmatrix} \tag{6}$$

where the target's relative position is $\vec{r}_{rel} = [r_x, r_y, r_z]^T$ and relative velocity is $\dot{\vec{r}}_{rel} = [\dot{r}_x, \dot{r}_y, \dot{r}_z]^T$.

In an Earth-centered frame, the relative state is the difference between the target state and the sensor state, provided in Equation 7.

$$\vec{x}_{rel} = \vec{x}_{target} - \vec{x}_{sensor} \tag{7}$$

The target's velocity in the image frame is needed to determine an appropriate exposure time. In order to determine the target's velocity in the image frame, the target's angular velocity must be known. The angular velocity of the target is given in Equation 8.

$$\vec{\omega}_{rel} = \frac{\vec{r}_{rel} \times \dot{\vec{r}}_{rel}}{\|\vec{r}_{rel}\|^2} \tag{8}$$

The speed tangential to the image frame is then given in Equation 9.

$$\dot{r}_{frame} = \|\vec{\omega} \times \vec{r}_{rel}\| \tag{9}$$

The tangential linear speed is converted to tangential angular speed by scaling the linear speed with the object's range, $\|\vec{r}_{rel}\|$, shown in Equation 10.

$$\omega_{frame} = \frac{\dot{r}_{frame}}{\|\vec{r}_{rel}\|} \tag{10}$$

The exposure time τ_{exp} is finally determined by dividing the smallest frame dimension by the tangential angular speed of the object relative to the observer, shown in Equation 11. A scaling factor is applied so that the streak does not span the entire frame. Because the object's predicted position inherently possesses some error, the exposure time must be scaled down to reduce the likelihood that the streak exits the frame.

$$\tau_{exp} = \frac{\beta * min(w,h)}{\omega_{frame}} \tag{11}$$

where β is a scaling factor for the streak size, w is the angular width of the image frame, and h is the angular height of the image frame. The current implementation sets $\beta = 0.3$, so the intended maximum extent of a streak is 30% of the frame. This setting for β was determined experimentally, largely through testing. Spanning 30% of the frame, streaks are still long enough to detect and characterize.

4. SAFETY MONITORING

4.1 Weather management

Access to information about weather conditions is critical for the system to determine if it is safe for the dome to open. The SSA Observatory monitors the instantaneous conditions in its immediate vicinity with a local weather station. The system also pulls forecast data from weather services to monitor the forecast. Specifics on these weather data sources are detailed in this subsection.

4.1.1 Local weather conditions

There are several commercial options for personal weather stations that are suitable for telescope operation. The SSA Observatory uses a Boltwood Cloud Sensor II to measure the local weather conditions. The Boltwood records several useful metrics, including temperature, wind speed, dew point, and precipitation. The measurements are written to a text file, which can be ingested by custom software for further analysis and action. In the current implementation, the local conditions are checked every five seconds.

Some conditions are safe for the equipment within specific bounds, while others are hazardous to the equipment if present at all. The bounded conditions are described in Table 2.

If any of the following conditions are detected, it is deemed unsafe for operation:

- precipitation
- wetness
- daylight

Table 2: Configured weather thresholds for Boltwood Cloud Sensor II

Condition	Minimum value	Maximum value
Ambient temperature (AT)	$0°C$	$37°C$
Wind speed	N/A	$15mph$
Relative humidity	N/A	40%
Dew point (DP)	N/A	$20°C$
$AT - DP$	$5°C$	N/A

It is important to check for wetness in conjunction with precipitation. Wetness could be caused by condensation or simply water remaining from recent precipitation; both are hazardous to computer hardware. While daylight poses no threat to the computers, direct sunlight into the telescope aperture may cause damage without proper filters equipped. We evaluated the performance of the local weather station against a regional weather station over the course of many weeks and found wind readings to be lower than expected and temperature readings to be higher than expected.

4.1.2 Forecast conditions

Forecasted weather conditions are freely available, courtesy of the National Weather Service. For the purpose of safety monitoring, the forecast is primarily checked for precipitation, though other metrics like wind speed and temperature can be monitored as well. In the current implementation, the forecast up to five hours into the future is analyzed – if a chance of precipitation $>25\%$ exists in the forecast, conditions are deemed unsafe.

Because the forecast is relatively unchanging over short timescales, the conditions in the forecast are checked only twice an hour.

4.2 Failure resiliency

Sohbrit must be resilient against power/network outages and hazardous weather conditions to operate autonomously and without supervision. Our approach tailored to the capabilities of the hardware components we have. Others have written about their dome safety mechanisms, including their approaches for weather monitoring [15], watchdogs [1], and cameras in and around the dome to monitor operations and sky conditions [14].

The observatory dome is an Astro Haven clamshell dome. The dome shell protects the equipment inside from wind, precipitation, sunlight, and extreme temperatures. When the dome is open, unfavorable weather conditions could damage the equipment. The dome automatically closes when the weather conditions become unsafe. The dome also closes when there are no active tasks, to mitigate the risk of exposing hardware to harmful conditions in the event of power loss.

The primary motivating scenario for developing dome safety mechanisms has been the open dome loses power and then a rainstorm moves in and drenches the equipment inside the dome. We have also considered scenarios involving network outages, crashed processes, and forgetting to close the dome after a visit.

While many scenarios can be handled automatically within the dome, some situations may require manual intervention for rectification. The clamshell dome does not have the ability to be manually closed and in the case of power loss, we are prepared to send operators to put a tarp over the open observatory. Network loss immediately isolates the observatory, and so all status information must be sent out of the dome preemptively to inform operators of the situation without being at the dome.

4.2.1 Approach

We use external database and monitoring tools to observe the state of the observatory and report potential dangers. We integrate with Elastic's Beats [5], Elasticsearch [6], and Kibana [7] tools to capture, store, and analyze system status. Custom analytics have also been developed for detecting scenarios involving more than one service or system.

The monitoring system, as depicted in Figure 3, uses the Beats modules Filebeat, Heartbeat, and Metricbeat actively gather logs, system status, and system metrics from across the system. Custom processes are used to send local

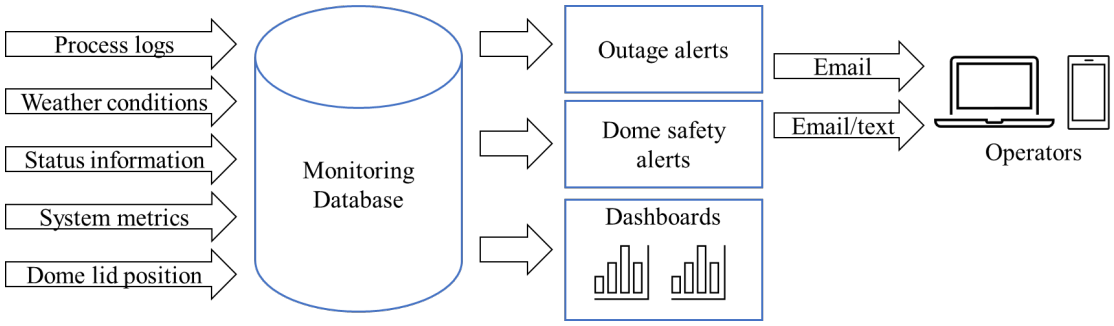


Fig. 3: Primary components of the monitoring system

weather data and data from the National Weather Service to the Elasticsearch database. Elasticsearch indexes all the data for later search and discovery. Kibana provides dashboards for viewing log volumes, system metrics, and system outages. Kibana also offers an alerting capability, which is configured to send a notification when certain systems are unreachable or logs from specific processes are no longer being received. We have also implemented a Kibana alert to notify us if the dome has been left open longer than allowed; the allowable threshold is currently set at 6 hours.

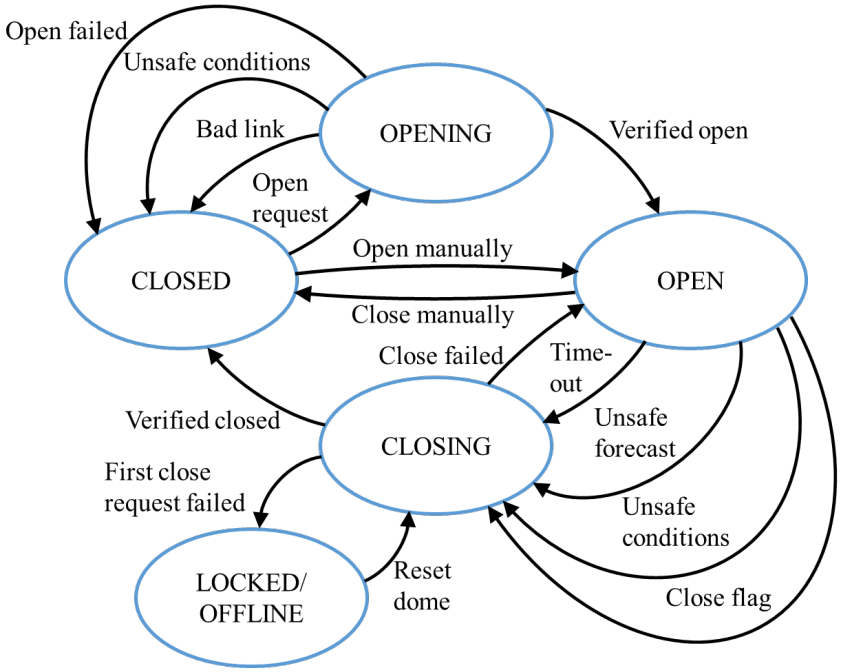


Fig. 4: State diagram for the dome lid position

The dome control software checks the local weather conditions and tests the link to the database before attempting to open the dome. A key principle is to never automatically open the dome without writing the intent do to so in the external database. Periodic weather checks, including a local weather conditions check every sixty seconds and a forecast check every thirty minutes, will close the dome if needed. When the dome is opened, a timeout is specified, and it will close at the end of that time unless if the close time is pushed out further. If the dome is not responding to a close request, then the software sends a reset command and tries again. These states and actions are captured in the state diagram in Figure 4.

When operators visit the observatory and take manual control of the dome lid, they are responsible for ensuring the telescopes do not point at the Sun and that they close the dome when they are finished. Operators cannot simply override the safety checks by using the buttons in the dome since the software will repeatedly attempt to close the dome. If an operator stops the software, then they will receive email notifications that it is no longer running.

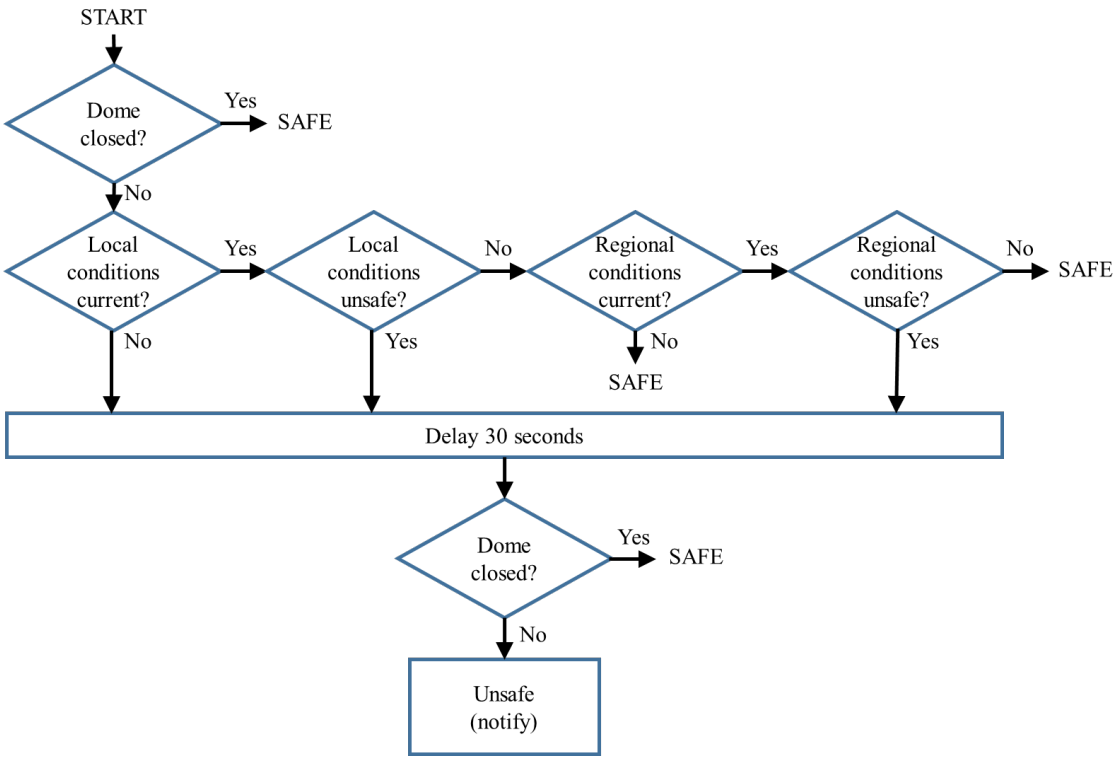


Fig. 5: Decision tree for dome safety notification

The dome safety checker runs on a machine outside of the observatory, and it analyzes the information in the database to detect unsafe situations (see Figure 5). It retrieves the latest dome position record and latest local and regional weather information from the database. For the case that the dome safety checker identifies a weather threat before the dome has the opportunity to close, a thirty-second wait is performed, and the dome status is checked again. If the dome is then closed, the system is considered safe. If the dome is not closed, or the state is unknown, then notifications are sent out over email and text messaging to the operators. The operators can then manually check the weather and system statuses and decide as to whether they need to rush out to the dome.

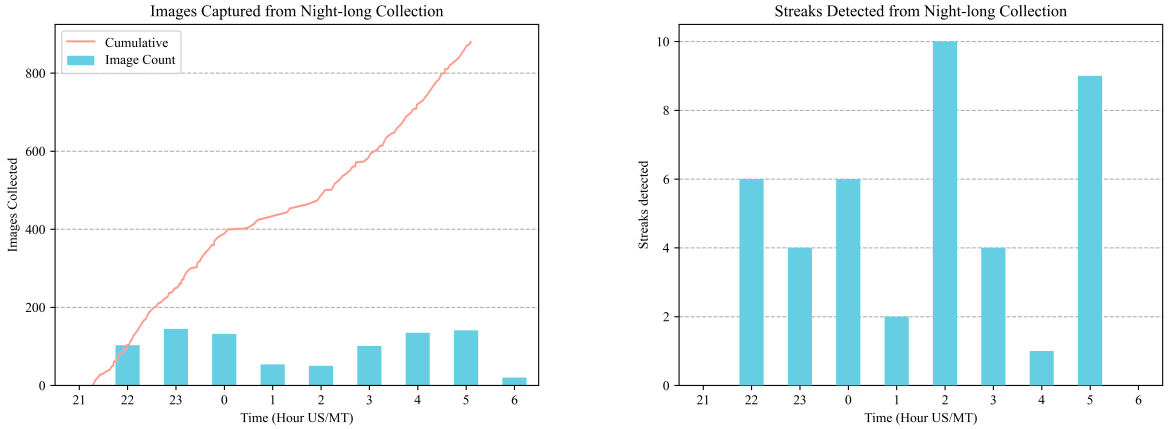
We have validated our monitoring and dome safety mechanisms through operator review. We have worked to reduce false positives to increase trust in the notifications that are sent out. We have also simulated various threat scenarios by cutting power/network and spraying water on the local weather station. A standard spray bottle works well at simulating an actual rainstorm with the Boltwood Cloud Sensor II.

5. DATA COLLECTION AND PROCESSING

During an exemplar autonomous collection on a clear night, Sohbrit captured 881 images and collected on 130 different targets. As the imagery is collected, it is stored on an internal cloud service, and available to a broad community of relevant data consumers for processing and analysis. When the data is pushed up to the cloud, a plate solve routine kicks off automatically for each image. The distribution of images collected throughout the night is shown in Figure 6a.

All images from the exemplar collection were processed separately in a streak detection routine, using the ASTRiDE Python package [11]. The streak detection software is still under development but is presently capable enough for processing. Some modifications have been made to ASTRiDE to reduce computation time – the modified streak detection process lasts about one second per image. Notably, some dim streaks were not detected by the software, but are visible to the human eye. It was estimated that the streak detection routine has a limiting magnitude of approximately 7.4, inferred from nearby stars in the frames of undetected streaks. Of the 881 images captured, 41

images contained streaks brighter than 7.4. The distribution of the software-detected streaks throughout the night is shown in Figure 6b.



(a) Distribution of images captured during collect

(b) Distribution of bright streaks captured during collect

Fig. 6: Captured images and detected streaks within during exemplar collect

A significant portion of the detected streaks were collected from the Starlink mega constellation. The Starlink satellites are especially bright just after sundown and just before sunrise, when the Sun angles are favorable for Earth-based observers. The favorable Sun angles are the primary reason behind the bifurcation of the image collections in Figure 6a – many resident space objects are visible in twilight, to the point that dome is open for hours at a time as Sohbrit collects images continuously. Conversely, near midnight there are fewer tasking opportunities, and the dome will close for periods when no visible objects are in view.

During nightly tests, it was noted that TLEs sometimes failed to accurately predict the trajectory of Starlinks, more so than other resident space objects. Residual errors were high enough to cause Starlinks to streak out of frame during exposure. It is suspected the unusual residual errors could be due to unmodeled station-keeping burns.

The comparatively large number of streaks detected at 01:00 in Figure 6b is due to a fortunate flyover of a GONETS M communications satellite. The GONETS M target was bright enough to be detectable by the streak detection algorithm for the entirety of the pass, so all ten images contained measurable streaks. In general, constellations with high inclinations and altitudes greater than LEO will have more observation opportunities throughout the night. Consequently, GONETS M satellite uniquely skewed the distribution of streak detections in this example, but a streak detection algorithm with a greater limiting magnitude would likely result in a bifurcated distribution of detections like Figure 6a.

Figure 7 shows an example of a bright streak that was detected by the streak processing software. The target, Oneweb-0646, was slightly obscured by cloud cover, which is the source of bright noise in the image. Encouragingly, the streak detection routine succeeded, even through the light cloud cover. Note the streak spans about 30% of the frame, consistent with the procedure outlined in Subsection 3.4.

Future work will seek to combine the plate solve and streak detection routines to automatically perform orbit determination. Additionally, the streak detection will be leveraged for closed-loop tracking, which motivated the minimization of computation time.

6. SUMMARY

Sandia National Laboratories' SSA Observatory is an automated SSA data collection system, capable of operating without a human-in-the-loop. Shown in this paper is hardware, software, and infrastructure design choices in support of unmanned data collection. A successful night of collection was shown, where over 800 images were collected between sundown and sunrise. The capability to acquire extensive nightly data will facilitate future efforts to develop

(a) Raw frame

(b) Processed frame

Fig. 7: Raw and processed frames of Oneweb-0646, taken with 80mm

advanced algorithms and maintain awareness in the space domain.

7. ACKNOWLEDGMENTS

This work was supported by the U.S. Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND2023-08628C.

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

8. REFERENCES

[1] Carl W Akerlof, RL Kehoe, TA McKay, ES Rykoff, DA Smith, DE Casperson, KE McGowan, WT Vestrand, PR Wozniak, JA Wren, et al. The rotse-iii robotic telescope system. *Publications of the Astronomical Society of the Pacific*, 115(803):132, 2003.

[2] Phil Bland, Greg Madsen, Matt Bold, Robert Howie, Ben Hartig, Trent Jansen-Sturgeon, James Mason, Dane McCormack, and Rod Drury. Fireopal: Toward a low-cost, global, coordinated network of optical sensors for ssa. 2018.

[3] Nicholas Paul Blazier, Samuel Tarin, Nathanael JK Brown, Prabal Nandy, and Drew P Woodbury. Sohbrit: Autonomous cots system for satellite characterization. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2017.

[4] Rita L Cognion. Large phase angle observations of geo satellites. In *Sensors and Systems for Space Applications VI*, volume 8739, pages 194–205. SPIE, 2013.

[5] Elastic. Beats. `https://www.elastic.co/beats`, 2023.

[6] Elastic. Elasticsearch. `https://www.elastic.co/elasticsearch`, 2023.

[7] Elastic. Kibana. `https://www.elastic.co/kibana`, 2023.

[8] Miguel Grinberg. *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.", 2018.

[9] Mark Hammond and Andy Robinson. *Python programming on win32: Help for windows programmers*. " O'Reilly Media, Inc.", 2000.

[10] J Hostetler and H Cowardin. Experimentally-derived phase function approximations in support of the orbital debris program office. In *International Orbital Debris Conference (IOC)*, number JSC-E-DAA-TN73676, 2019.

[11] Dae-Won Kim. ASTRiDE: Automated Streak Detection for Astronomical Images, May 2016.

[12] Mark Mulrooney, Mark J Matney, Matthew D Hejduk, and Edwin S Barker. An investigation of global albedo values. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, pages 16–19. Curran Associates, Inc. Redhook, NY, 2008.

[13] Brandon Rhodes. Skyfield: High precision research-grade positions for planets and earth satellites generator. *Astrophysics Source Code Library*, pages ascl–1907, 2019.

[14] Adam B Smith, Daniel B Caton, and R Lee Hawkins. Implementation and operation of a robotic telescope on skynet. *Publications of the Astronomical Society of the Pacific*, 128(963):055002, 2016.

[15] KG Strassmeier, T Granzer, M Weber, M Woche, MI Andersen, J Bartus, S-M Bauer, F Dionies, E Popow, T Fechner, et al. The stella robotic observatory. *Astronomische Nachrichten: Astronomical Notes*, 325(6-8):527–532, 2004.

[16] United States Government Accountability Office. Space situation awareness: DOD should evaluate how it can use commercial data, April 2023.

[17] David A Vallado. *Fundamentals of astrodynamics and applications*. Microcosm Press, fourth edition, 2013.