REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
05/01/2020	Technical Report	

4. TITLE AND SUBTITLE
Sensor Network, Autonomous, Resilient, Extensible (SNARE): Logic and Application Comparison to Special Perturbation (SP) Tasker

5a. CONTRACT NUMBER
FA8702-20-C-0001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Burchett, Dustin R.; Carden, Robert D.;

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420

8. PERFORMING ORGANIZATION REPORT NUMBER
PRS-20-1509

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Space Command / Space and Missile Systems Center

10. SPONSOR/MONITOR'S ACRONYM(S)
SMC / SPG

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This paper proposes a mathematical basis for a collection system resource management method that promises to provide near real time persistent awareness and enables the tenets and objectives of SPD-3, and SPD-4. Implemented properly, this methodology could become the logical basis for a functional architecture that unifies the system for the coordination of sensing capability to maintain near real-time positional awareness and change detection on tactically relevant time lines.

15. SUBJECT TERMS
Sensing and Signal Processing (General); Space Operations; Resident Space Object; Space Sensor Network; SNARE; SSN; Sensor Network Autonomous Resilient Extensible; RSO;

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Susan Carpenito
				76	**19b. TELEPHONE NUMBER** *(Include area code)* 781-271-7646

MP200316
MITRE PRODUCT

MITRE

Sensor Network, Autonomous, Resilient, Extensible (SNARE)

Logic and Application Comparison to Special Perturbation (SP) Tasker

Dustin Burchett
Robert Carden
May 2020

Sponsor: Gordon Kordyak, SMC/SPG
Dept. No.: P433 - Space Domain Awareness & C2
Contract No.: FA8702-20-C-0001
Project No.: 03208SH0-0A

Colorado Springs, CO

This page intentionally left blank.

Abstract

Scheduling observations on the current Resident Space Object (RSO) space catalog to the Space Sensor Network (SSN) is a resource-constrained optimization problem. Bound by observation and revisit requirements, known scheduling violations can be minimized under the resource constraint using a selective decision process. Assigning values of collection to determine which sensor collects which observations over time is the initial path to optimizing this problem. This decision process utilizes properties of the individuals within an RSO catalog as well as properties of the system as a whole such as sensor work-load distribution and sensor-RSO visibility. Each sensor-RSO pairing receives a value of collect which can be easily compared against other sensor-RSO pairings to help determine an optimal collection scheme.

This page intentionally left blank.

Table of Contents

List of Figures

List of Tables

This page intentionally left blank.

1 Introduction

U.S. National Interest has become critically dependent on space capabilities for nearly every aspect of modern daily life. Communication, Positioning, Navigation, and Timing (PNT), tracking, weather, agriculture . . . all rely on space-based capabilities that require the freedom to operate in space. Historically the high cost of entry has limited the space domain to nation states with large budgets. That has changed dramatically in the last 10 years; payloads are getting smaller and cheaper to build. Launch service costs are falling dramatically as more and more companies providing lift develop capacity. In 2018, the cost of launch and payload development had dropped to the point where it was feasible for a high school[1], several colleges[2] and even an elementary school[3] to design an experiment, secure lift, and place their satellites in orbit.

Cost as a barrier to space development has fundamentally been removed.

In the near future, nearly unlimited access to the domain will complicate freedom to operate. The future sustainability of what is predicted to be a $558 billion[4] industry by 2026 will require effective global management of the domain. Long-term Sustainability (LTS) must be considered by every actor in space, considering industry plans for nearly 15000 operational satellites to be launched between 2017 and 2026[5]. The United Nations Committee on Peaceful Uses of Outer Space (UN COPUOS) identifies the number one risk to space development as space debris. It identifies the second as the large predicted number of satellites. The convergence of these two risks will drive the need for management of the interaction between each Resident Space Object (RSO), debris and operational, or more colloquially Space Traffic Management (STM).

Congestion of the domain requires active cooperative management to ensure freedom to operate.

There are many recommendations as to what is needed to achieve STM from legal frameworks[6] to operational guidelines from the UN[7]. But at their core, real-time positional knowledge is the enabler, i.e. where things are in space and time. The tenets of U.S. National Space Strategy and global viability of the domain require the ability to know where objects are, where they are going, and their relationships to other objects. Today, that knowledge is primarily based on collections from a core group of eight dedicated and eighteen multiple-mission sensors, primarily operated by the Department of Defense (DoD). Space Policy Directive (SPD), signed into policy by President Donald J. Trump[8], recognizes that this infrastructure and the previously military methods of STM will become inadequate as space continues to become more crowded and contested.

This paper proposes a mathematical basis for a collection system resource management method that promises to provide near real time persistent awareness and enables the tenets and objectives of SPD-3[8], and SPD-4[9]. Implemented properly, this methodology could become the logical basis for a functional architecture that unifies the system for the coordination of sensing capability to maintain near real-time positional awareness and change detection on tactically relevant time lines.

2 Problem Definition

The motivation behind Sensor Network, Autonomous, Resilient, Extensible (SNARE) is to achieve two goals: Accurate awareness of the positional state of objects and tactically relevant persistence for change detection of space objects. Combined, these two ideas are the positional basis for modern Space Domain Awareness (SDA), formerly Situational Space Awareness (SSA), and STM. In the congested near future of space, these two functions must be automated and able to scale with the expected quantity of objects. Currently, there is no automated process that utilizes sensing resources to achieve both objectives. In addition, it is likely the overall accuracy of the catalog can be improved through an automated process.

These motivations contrast the intent of the system in use today. The Special Perturbation (SP) Tasker computer program was designed to task the Space Sensor Network (SSN) with observations to maintain an accurate satellite catalog[10]. SP Tasker generates daily tasking files which are aggregated to the composite tasking file. The composite tasking file is loaded into the Space Defense Operations Center (SPADOC) where it and Element Set (ELSET) messages are sent to the SSN sensors. SPADOC generates the Composite Tasking List (CTL) directing how many tracks and observations sensors need to collect over the next 24-hour period[11]. The ELSET provides the latest information on the location of the satellite. Each SSN sensor receives a subset of the CTL containing only the satellites that are tasked to that sensor. As sensors execute their taskings, the collected observations are sent back to SPADOC where they are used to update the ELSETs which are sent to SP Tasker for consideration in the next day's tasking. Additionally, SP Tasker utilizes sensor performance metrics, satellite energy dissipation rates, and radar cross section/visual magnitude of satellites to align sensor/satellite parings as it generates the next day's tasking file[12].

Because SP Tasker sends general tasks for an entire 24-hour period, it does not support tactical operations which might require decisions in mere minutes. And, because reporting object collects are not always processed in real time, it can take hours or even multiple days to acquire enough evidence of a potentially nefarious object's intent or activities. When an RSO eventually becomes interesting due to such activity, manual overrides on the existing schedule must occur in an attempt to recover the object with the desired positional accuracy. In the sense of space warfare, these timelines of recovery do not support the threat assessments or space flight safety missions.

2.1 Goal

This paper provides a decision process framework that can lead to a collection scheme across an enterprise of sensors, not all of which are controlled by the U.S. To retain context, this framework should be sensor agnostic, extensible (even beyond the SSN), real-time applicable, and meet or exceed guidelines and requirements set forth by the current tasking scheme. The framework should then demonstrate that solutions exist for achieving both persistent awareness of objects in an ever-growing catalog and also tactical (real-time) change detection, a capability which is not considered in the current collection paradigm. The system that is in place today meets the current requirements for catalog-keeping, but the capabilities of SNARE could eventually set the bar for stricter requirements or serve as a reference architecture if stricter requirements emerge due to modern operational needs.

In particular, the aim of sharing the SNARE logic is to mathematically demonstrate the following key concepts:

- Catalog accuracy and change detection can be maintained through a simple decision process.
- A decision process can be reached through the application of a utility function, which assigns instantaneous value of each possible collect and in turn forms a collection scheme.
- As available resources scale, observable faults that contribute to a sub-optimal collection scheme become minimized.

To contrast with the methodologies put into place today, it is also desirable to explore the following ideas about a collection scheme:

- Is there a way to avoid duplication of work?
- Can sensors be utilized ambiguously to contribute to the collection scheme?
- Is it possible that the SNARE concept can free up network sensing resources while maintaining acceptable accuracy across the majority of the catalog?
- If each of the above is met, can a scheme be built in real time?

2.2 Metrics

A chosen selection scheme needs to be considered against certain metrics to explain not only *if* it is useful, but *how* it might be useful. Consider a few basic measures of what could indicate a good collection scheme.

- Diversity—how many unique objects are being tracked in a given scheme? The measure of diversity in a 24-hour scheme is referred to as dv, where having diversity is valuable.
- Redundancy—how many sensors are being told to do the same work at the same time? A penalty in redundant tasking (not collection) results in a multiview violation, mv. Note that some missions could require triangulation or multiple simultaneous collections.
- Revisit Expectation—are objects being seen at the appropriate rate relative to their interest level to be able to detect changes in its orbit? Objects that miss their revisit time in the collection scheme incur revisit violations, rv.
- Priority Expectation—are less important objects often being tracked more than other, more important, trackable objects? Each object that is in the collection scheme that replaces a more important visible object commits a priority violation, pv.
- Observation Expectation—are objects being tracked enough to maintain or improve catalog accuracy? Objects that are lowering the catalog accuracy due to insufficient tracking cause observation violations, ov.

In addition to these metrics which are based off a measure of diversity and counting violations in a collection scheme (unitless), consider two other metrics that are closely related but already have meaning and use outside of the SNARE construct.

- Total Violation Time (TVT)—How much time (in minutes) between collections on an object versus its requirement?
- Total Positional Error—What is the total expected positional error (kilometers) of objects in the catalog, due to track rate?

These topics will be more formally defined and discussed in Section 4. First, the problem notation will be explored.

2.3 Sensor Notation

Let $\sigma_i, i \in \{1, \ldots, n\}$ denote some i^{th} sensor in the SSN (out of n sensors). Let there be t timesteps in a day, and let timestep $j = 1 + (T - 1) \bmod t$, where $T \in \mathbb{N}$ is the number of timesteps from $T_0 = 0$. Note that as T is the number of timesteps after T_0, then $T > 0$, and hence $j \in [1, t]$.

- Let $C = \sum_i c_i$ be the theoretical track capacity for the SSN in a single timestep, where c_i is the theoretical track track capacity for a single σ_i in a timestep, and $D = \sum_j C$ denote the daily track capacity for the SSN

There is currently no restriction on the actual size of timestep j. Operationally, j could relate to however fast information could be processed at a sensor level.

2.4 RSO Notation

Let $\psi_k, k \in \{1, \ldots, m\}$ denote some k^{th} RSO (out of m objects). Each ψ_k will have several properties that are either determined or monitored.

- Let p_k describe the interest level, or priority level, of ψ_k such that $p_k = k$. The lower the value of the priority level, the higher the interest is in the object (note that the priority level of an object is unique and ordered)
- Define $r_k(j)$ as the count of timesteps from time j since ψ_k was last seen. In practice this quantity is only considered the timestep after the collect occurred, and hence $r_k \geq 1$. For example, assume that an object ψ_k hasn't been observed at $j = T_0 = 0$. Then, $r_k(1) = 1$. Then, ψ_k is seen at $j = 10$. Due to the nature of when this step is performed, $r_k(10) = 10$ but $r_k(11) = 1$.
- Let $\hat{r}_k$ be the maximum allowable number of timesteps between tracks on ψ_k before a revisit violation

- Define $o_k(j)$ as the count at timestep j over a given collection in which ψ_k has been observed, where $o_k \geq 0$. Note that this should not be confused with the empirical collection of actual formatted *observations* that might be sent between a network of sensors. In fact, it is assumed within the nature of this work that a "completed observation" as defined by the observation count is actually a successful track.

- Let $\hat{o}_k$ be the required number of tracks on ψ_k in t timesteps to avoid an observation violation

2.5 Collection Notation

Denote the proposed collection scheme for a given timestep j to be the multiset $\dot{\mathcal{S}}_j \in \mathbb{Z}^{C\times 1}$, and the verified schedule over the last t timesteps at time T be denoted as the multiset $\mathcal{S}_T \in \mathbb{Z}^{C\times t}$, where $\mathcal{S}_T = \{\mathcal{S}_j\}_1^t$. It follows that $o_k(j) = \sum_j |\{k \in \mathcal{S}_j\}|$.

- Let $v_{i,k}(j)$ describe the visibility of ψ_k by σ_i at time j. A basic consideration of this value is binary (where a value of 1 means it is visible and a value of 0 means it is not visible), however could be extended to any number in the range of $[0, 1]$ for various interpretations of "visibility".

3 Model Flow

The proposed collection scheme-building paradigm follows a simple data flow model. The concept extends from the idea of simplicity; given a small amount of information at a given time, can it be utilized to structure a tasking decision for how the network responds to it? In particular, given a published state of ψ_k at time j, what information can be utilized in the construction of a collection scheme for each σ_i?

3.1 Decision Process

Parameters $p, \hat{r}, \hat{o}$ are defined using government policy, orbital analyst tools, and potentially Capstone Requirements Document (CRD) standards. Assume visibility between sensors and objects is relatively understood within a small time frame. A utility function can then use known qualities and object observation history in order to quantify relative value between tracks at a given point in that same time frame. Once values between sensors and RSOs are established, each sensor can make an informed decision on how they should be filling their duty cycle in the next time epoch. A proposed schedule can be distributed between sensors, where the sensors can then optimize their own task list to achieve the highest probability of capturing each proposed track, and required set of observations on the track.

3.2 Using a Utility Function

The purpose of the utility function is to help quantify the relative value of a collect on an object by using the known information about the object. In general, the utility function should have a few objective qualities.

- A higher priority object should correlate to a higher value of collect. To avoid only high-ranked objects from being consistently tasked, there needs to be other ways for lower interest objects to acquire value.
- Value of collect on an object increases as separation increases from its last collect. In particular, there should be increase in value as an object approaches its revisit expectation to avoid a violation. If an object surpasses its revisit expectation, the rate of which its value increases should change.
- Value of a collect on an object is higher when it hasn't been seen yet in its collection period. Once an object has been seen enough times to maintain defined accuracy on the object, the value of a collect drops considerably.
- Value of multiple collects on a single object at the same time is the same as a single collect on that object at a given time, unless specifically defined otherwise.
- All valid collections add some utility to the network.
- If a sensor-object pairing is not possible (i.e. an object is outside of a sensor's field of regard), the collection is invalid and the utility of collect is zero.

It can now be said that the utility function can be generally defined as some function L to find utility, U, such that

$$U = L(v, r, \hat{r}, o, \hat{o}, p)$$

As such, the given inputs of L are differentiated, and if each are independent from each other, U can be reformulated.

$$U = v * f(r; \hat{r}) * g(o; \hat{o}) * h(p),$$

for some functions f, g, h. An example of a selection of functions f, g are included later in the text, Section 5.2.

In practice it is certainly possible that there is some co-dependence between some of the inputs, i.e. r might correlate with p. Even if this is the case the analysis will be the same. To define in context, subscript notation can now be used to deepen the relationship between σ_i and ψ_k in a discretized time system,

$$U_{i,k}(j) = v_{i,k}(j) * f(r_k(j); \hat{r}_k) * g(o_k(j); \hat{o}_k) * h(p_k)$$

An additional layer to the utility function to account for reactiveness of the system will be referred to as the "bounty list". Let, the bounty list, ℓ, be defined as a transformation T on U,

$$T(U) \to U$$

$$\text{where } T(U) = \begin{cases} U^{\alpha}, & \forall k \in \ell \\ U, & \forall k \notin \ell \end{cases} \text{ and } \ell = \{k : k \in \dot{\mathcal{S}}_j \backslash \mathcal{S}_j\}$$

for $\alpha > 1$. Objects that are tasked that don't get collected are placed on the bounty list. Larger values for α will raise the value of collect for bounty list objects much quicker than an α close to one.

The primary considerations in this scheme are the revisit and observation properties of each RSO. It is likely that in application priority level will correlate appropriately between the revisit and observation expectations, and hence should not hold a lot of weight in determining utility of collect. However, priority will be used as a "tie-breaker" (in case multiple objects have the same utility of collect at a given time). Before a particular utility function is addressed, assume that the above properties are met and the above utility function will be used as a place holder when describing the model flow.

3.3 Base Case

Consider the interaction between a catalog of a single object, ψ_1 and a single sensor, σ_1. In this case $p = 1$, and $\hat{r}$ and $\hat{o}$ can be chosen arbitrarily. Sensor capacity, c_1, can be arbitrary in this example. The pseudocode is provided below:

- $T = T_0$
- $r_1 = 1$, and as $\mathcal{S}_T$ is empty, $o_1 = 0$
- σ_1 reports when it can next see ψ_1

  - If $v_{1,1} = 0$ then $U = 0$, and there are no objects to put into $\dot{\mathcal{S}}_{1,1}$ and it remains empty
  - If $v_{1,1} > 0$ then f, g are calculated. Hence, $U = f * g$, and $\dot{\mathcal{S}}_{1,1} = \{1\}$

- T, j increment
  - If $\dot{\mathcal{S}}_{1,j}$ becomes a verified collection, $\dot{\mathcal{S}}_{1,j} \rightarrow \mathcal{S}_{1,j}$ (which in the base case becomes the first and only element of $\mathcal{S}_T$). Then $r_1(j) = 1$ and $o_1(j) = 1$, so f stays the same, and g decreases
  - If ψ_1 was missed by σ_1 then $r = 2$ and $o = 0$, which means f increases but g stays the same
- Visibility for the next time step is gathered, and the utility function is updated
- A new schedule, $\dot{\mathcal{S}}_{1,j+1}$, is proposed and executed
- T, j increment and process from $T > T_0$ repeats

A flow chart is provided in Section 11, Figure 20.

3.4 Extended Case

The above data flow model is similar to the extended case when vectorized with $n > 1$ sensors and $m > 1$ objects. Instead of a single utility of collect, $U \in \mathbb{R}$, the utility function returns a matrix of information, where $U \in \mathbb{R}^{n \times m}$. Note that the visibility component can increase the complexity of the value function if $v_{i,k}$ is unique from sensor to sensor. To keep things simple, let's assume there is a decision process by each sensor that verifies whether an object is visible for collection or not, but inevitably there is no effective difference between any one collect in regards to the utility calculation (in other words, $v \in \{0, 1\}$). Other assumptions about things like sensor duty cycle and theoretical sensor capacity could be lumped into the visibility portion of the utility function to give it range, but for now it will be referred to as a binary value. As there is no logical restriction on two separate objects sharing similar $\hat{r}$ and $\hat{o}$ values while varying p, distinct values of U for varying p is desired. So in this case, $U \in \mathbb{R}^m$, where

$$U_k(j) = f(r_k(j); \hat{r}_k) * g(o_k(j); \hat{o}_k) * h(p_k)$$

and each sensor either can see ψ_k and has $U > 0$ utility on collect, or cannot see ψ_k and thus has $U = 0$ utility of collect at time j.

Consider the interaction between a catalog of objects $\{\psi_k\}_1^m$ and a network of sensors $\{\sigma_i\}_1^n$. Recall that, for simplicity, $p_k = k, \forall \psi_k$. Let $\hat{r}_k$ be arbitrarily correlated with p_k.

In the extended case we will also introduce the idea of "sensor regrets." In application, a task list can be sent to a sensor (or even self-determined) and it will respond with either an affirmation (i.e. the requested task list can be accomplished by the sensor), or a regret list. The regret list is the set of tasks that the sensor has to give up in order to accomplish the given task list.

- $T = T_0$
- $\{r_k\}_1^m = 1$, and as $\mathcal{S}_T$ is empty, $\{o_k\}_1^m = 0$
- σ update $v, \forall k$ with *next visible time only*
- Bounty list, ℓ, is initialized as an empty set
- For $v_{i,k} > 0$, f, g components are calculated in the utility function
- While $\sum_{i,k} U > 0$...
  - $K = \max_{i,k} U$ is selected
  - If σ_i has no regrets for taking K, it is absorbed by $\dot{\mathcal{S}}_{i,j}$ and $U_k = 0, \forall i$
  - If σ_i has regrets for taking K, then $U_{i,k} = 0$
  - If σ_i reaches capacity, c_i, then $U_i = 0, \forall k$
- T, j increment
- Verified collections from $\dot{\mathcal{S}}_{i,j} \rightarrow \mathcal{S}_{i,j}$ and populate $\mathcal{S}_T$
- A query on the next time $\forall k \in \mathcal{S}_j$ are visible is solicited from each sensor, and v is updated
- $r_k(j) = 1, \forall k \in \mathcal{S}_j$ and $o_k(j) = \sum_j \mathcal{S}_T | (\mathcal{S}_T = k), \forall k$
- $r_k(j)$ increment $\forall k \notin \mathcal{S}_j$
- Bounty list, ℓ is updated
- Utility is calculated, for all visible k at time j
- Utility is adjusted using the bounty list
- While $\sum_{i,k} U > 0$...
  - $K = \max_{i,k} U$ is selected
  - If there is a tie between the value between two sensors, the one that has more of its expected capacity left is used
  - If σ_i has no regrets for taking K, it is absorbed by $\dot{\mathcal{S}}_{i,j}$ and $U_k = 0, \forall i$
  - If σ_i has regrets for taking K, then $U_{i,k} = 0$
  - If σ_i reaches capacity, c_i, then $U_i = 0, \forall k$
- A new schedule, $\dot{\mathcal{S}}_{j+1}$, is proposed and executed
- T, j increment and process from $T > T_0$ repeats

A basic summary of the pseudocode is listed as follows:

1. All visible objects at each timestep are ordered by value of collect using the utility function.
2. The sensor with the highest value of collect (or most available duty cycle during a tie) gets tasked with that object.
3. Objects are removed from the list as they are tasked (under the impression the collect will be successful).
4. All objects are tasked with the maximum sized-list of collections possible, up to sensor capacity (when a sensor cannot collect anything else without regrets).
5. All collections that fail to occur are given more value on the next iteration.

A flow chart is provided in Section 11, Figure 21. The process of calculating $U_{i,k}$, then performing the translation $T(U)$, and then sorting and filtering U to decide what goes into $\dot{\mathcal{S}}_j$ will now be referred to as the Local Value Function (LVF).

4 Using a Global Value Function

Metrics on the decision process can be captured during or after the scheme-building process. The general tactic is to count violations as they occur, monitor them during the scheme-building process, and then tally them against each other in sets of "scores".

The scoring method will be described using the term Global Value Function (GVF). It can be generally defined as some function G used to compute some global value GV_T over a collection scheme $\mathcal{S}_T$, where

$$GV_T = G(\mathcal{S}_T)$$

The violation metrics were expressed earlier, but now they will be examined a little more closely. Recall the metrics, dv, mv, rv, pv, ov. Each metric differently describes faults in the schedule $\mathcal{S}_T$, and are the driving forces in the GVF. Hence, G can be expanded to the forces measuring $\mathcal{S}_T$ into five independent components.

$$GV_T = G(dv_T, mv_T, rv_T, pv_T, ov_T)$$

With a scoring function, it is important to be able to arrive at conclusions that can compare one instance of a scheme to another. Although there may be no absolute way to do this without first having definitive measures on every possible permutation on every type of schedule, a sense of relativistic value may be achieved through the process of normalization. In other words, if there is a way to compute a score of "goodness" on one metric with one set-up for a collection scheme, it may hold value to be compared versus the same metric on a completely different collection scheme (like getting an 88 score out of 95 on one exam in one class, and then a 92 out of 106 on homework in a different class). In particular, if each metric is normalized to a possible value from $[0, 1]$ out of 1, it might be useful in future analysis (however, it is also sometimes relevant to look at the individual counts of each violation rather than the score).

The general format for a scoring function is

$$\widetilde{\text{score}} = 1 - \frac{\sum \text{violations}}{\text{max violations possible}}$$

where $\widetilde{\text{score}} \in [0, 1]$. In this context, a score of "1" would be a perfect score. Note that within this general formulation, all violations that meet the criteria to be counted within a score set are treated equivalently.

4.1 Diversity Score

The diversity metric, dv is simply the count of unique ψ out of m total objects in schedule $\mathcal{S}_T$, and is denoted as

$$\widetilde{dv}_T = 1 - \frac{|\mathcal{D}_T|}{m} \text{ where set } \mathcal{D} = \{k : k \notin \mathcal{S}\}$$

The diversity metric is particularly useful in determining how greedy a collection scheme is. A greedy collection scheme will cap its diversity over time, depending on how much sensing resources

are available. It's important to note that additional collects on an already collected object will not add to this portion of the score, only collects on objects not seen within t timesteps will increase diversity.

The biggest set $\mathcal{D}$ can be is m, since that is the maximum number of objects observable occurring when the schedule is empty, and the smallest $\mathcal{D}$ can be is 0, assuming the schedule is populated with every object in the catalog. Hence, $\widetilde{dv}_T \in [0, 1]$.

4.2 Multiview (Redundancy) Score

The multiview metric, $mv_T(j)$ is the total count of duplicate k within each partition of $\mathcal{S}_T$ at time j. This metric monitors duplicate, or redundant work in a tasking scheme but does not penalize for collects that occur without being tasked to do so (i.e. there could be multiple instances of an object in $\mathcal{S}$ but there may not be any multiview violations). Hence, multiview violations can be defined as

$$mv_T = \sum_j \text{Supp}(\dot{\mathcal{S}}_j/\{0\}) - |\mathcal{D}_j| + m$$

The maximum multiview violations that can occur is speculative. If capacity measures are known for each sensor, then the maximum size of the schedule is known which gives an indication of how many possible multiview violations can occur. Assuming every sensor tracks the exact same object (and only that object), at every timestep, then $\max mv_T = t * (C - 1)$. In practice this quantity will be sufficient for an assumed C. However if $C \gg m$ it could be the case that $\widetilde{dv}_j = 1$, and yet $mv_j > 1$. This case also changes the nature of the assumed "resource constrained" environment, and hence will not be considered further.

Given the maximum possible value of mv_T, let

$$\widetilde{mv}_T = 1 - \frac{\sum_j \text{Supp}(\dot{\mathcal{S}}_j/\{0\}) - |\mathcal{D}_j| + m}{t * (C - 1)}$$

and under constrained capacity assumptions, $\widetilde{mv}_T \in [0, 1]$.

4.3 Revisit Score

The revisit metric, $rv_T(j)$ is the total count of timesteps that an object has been in violation of its revisit expectation, for all objects, at time j. The revisit count plays a crucial part in monitoring whether objects, particularly those that are of high interest, are being tracked enough in the sensor network. A revisit violation occurs when, for a given ψ_k, the revisit count at time j for ψ_k, $r_k(j) > \hat{r}_k$. Hence revisit violations can be defined as

$$rv_T = \sum_k \max\left(r_k - \hat{r}_k, 0\right)$$

Note that the maximum number of violations that can occur can grow to infinity as time progresses, as it is not cyclic like some of the other metrics. At any time T, a maximum count of revisit

violations can be bounded up to that point. Assuming that no objects are being tracked at all, and that $T \geq \max_k r_k$, then

$$\begin{aligned}\max rv_T &= \sum_k (r_k - \hat{r}_k) \\ &= \sum_k r_k - \sum_k \hat{r}_k \\ &= m * (T - \min_k r_k) - \sum_k \hat{r}_k\end{aligned}$$

However, recall that by construction $r_k \geq 1, \forall k$. Then, there cannot be any revisit violations at $T = 1$. Hence,

$$\max rv_T = m * (T - 1) - \sum_k \hat{r}_k$$

which can be formulated into a similar scoring metric as the previous,

$$\widetilde{rv}_T = 1 - \frac{\sum_k \max(r_k - \hat{r}_k, 0)}{m * (T - 1) - \sum_k \hat{r}_k}$$

where $\widetilde{rv}_T \in [0, 1]$.

4.4 Priority Score

The prioritization metric is unique in that it depends on whether another violation has occurred. In particular, a priority violation cannot occur unless a revisit violation has occurred first. If an object is visible in the network and a revisit violation occurs, then each object that is less important in the proposed schedule at time j counts towards the priority violation count. First, consider the set

$$\mathcal{P}_k = \{e : e > k \in \mathcal{S}_j\}$$

Then, assuming that there is a revisit violation on ψ_k at time j (or in other words, $r_k(j) > \hat{r}_k(j)$), and that at least one sensor could have seen it at time j ($\exists i : v_{ik} > 0$),

$$pv_k(j) = |\mathcal{P}_k|$$

and $pv_k(j) = 0$ otherwise. That conditional quantity can be written as

$$\min\left(\sum_i v_{ik} * \max(r_k - \hat{r}_k, 0), 1\right) \in [0, 1],$$

where the quantity is 0 when ψ_k cannot be seen by any σ_i *or* $r_k \leq \hat{r}_k$ and the quantity is 1 otherwise. As such,

$$pv_T = \sum_j \sum_k \min\left(\sum_i v_{ik} * \max(r_k - \hat{r}_k, 0), 1\right) * |\mathcal{P}_k|$$

The minimum number of priority violations occurs when there is either no visibility on an object by any sensor and/or there are no revisit violations. In which case

$$\begin{aligned}\min pv_T &= \sum_j \sum_k 0 * |\mathcal{P}_k| \\ &= 0\end{aligned}$$

Note that just because a revisit violation occurs, doesn't mean there has to be a priority violation. In general, a greedy algorithm might have zero priority violations but yet could still have a very low revisit expectation score. However, if there are no revisit violations a priority violation cannot occur so minimizing revisit violations in turn minimizes priority violations.

Now the maximum count of priority violations will be found. Assume that there is at least one sensor that can see every object at every timestep, then $\sum_i v_{i,k} \geq 1$. In addition, assume that $T \geq \max \hat{r}_k + t$ (so that there are maximum revisit violations assumed at every timestep in $\mathcal{S}_T$) and that $\mathcal{S}_T$ contains C objects, with unique elements $p \in \{m - C, \ldots, m\}\ \forall j$. Then,

$$\begin{aligned} \max pv_T &= \sum_j \sum_k 1 * |\mathcal{P}_k| \\ &= \sum_j \sum_k |\mathcal{P}_k| \end{aligned}$$

$|\mathcal{P}_k| = C$ at every timestep, by construction. Thus,

$$\begin{aligned} \max pv_T &= \sum_j \sum_{k=1}^{m-C} C \\ &= t * \sum_{k=1}^{m-C} C \\ &= C * t * (m - C) \end{aligned}$$

This finally arrives at the scoring metric, defined as

$$\widetilde{pv}_T = 1 - \frac{\sum_j \sum_k \min\left(\sum_i v_{i,k} * \max\left(r_k - \hat{r}_k, 0\right), 1\right) * |\mathcal{P}_k|}{C * t * (m - C)}$$

where $\widetilde{pv}_T \in [0, 1]$.

4.5 Observation Score

The observation metric has already been defined in context. Recall that $o_k(j) = \sum_j \mathcal{S}_T|(\mathcal{S}_T = k)$. The number of violations that occur in a scheme $\mathcal{S}_T$ is the count of tracks needed in order to avoid any violations. Hence,

$$ov_T = \sum_k \max(\hat{o}_k - o_k(j), 0)$$

Note that the max number of violations that can occur are when $o_k = 0, \forall k$. Hence, $\max ov_T = \sum_k \hat{o}_k$ and the observation score is defined as

$$\widetilde{ov}_T = 1 - \frac{\sum_k \max(\hat{o}_k - o_k(j), 0)}{\sum_k \hat{o}_k}$$

where $\widetilde{ov}_T \in [0, 1]$.

4.6 Additional Metrics

The other two metrics mentioned earlier will be defined here. These metrics are used in application outside of the SNARE construct.

4.6.1 Violation Time and Total Violation Time

Let Violation Time (VT) describe the amount of time surpassed between an object's collect and its revisit expectation. Then the current VT of ψ_k at time j is,

$$VT_k(j) = r_k(j) - \hat{r}_k, \forall k \in \mathcal{S}_j$$

where $VT \in [1 - \hat{r}_k, \infty)$. In practice when this quantity is positive the object is in violation of its expectation. Let $\mathcal{J}_k$ be the set of times at which ψ_k is collected while violating its expectation. Then let

$$\mathcal{J}_k = \{j | r_k(j+1) = 1 \wedge r_k(j) > \hat{r}_k\}$$

and the Total Violation Time for an object is then

$$TVT_k = \sum_{j \in \mathcal{J}_k} r_k(j).$$

The relationship between VT and TVT should now be clear. In relation to the SNARE construct, the goal is for every object to avoid committing this violation. However, to maintain persistent awareness on an object it is also important to minimize the variance between independent VTs as well as bring the average VT on each collect to zero or less.

4.6.2 Total Positional Error

There are many high-fidelity methods of analyzing positional error. However, positional error and accuracy confidence can be easily estimated using Space Surveillance Network Optimization #2 (SSNOII) curves and tables. The SSNOII table requires six inputs and then outputs estimated orbital accuracy (in kilometers). The six inputs are as follows:

1. Orbital regime—Geo-stationary Earth Orbit (GEO), Low Earth Orbit (LEO), Medium Earth Orbit (MEO), Highly Elliptical Orbit (HEO)
2. Energy Dissipation Rate (EDR) Group (using EDR bins)
3. Track mix type (the sensors in the simulation are a mix of radar and ground-based opticals and it is assumed that each sensor has the ability to collect on each object, although the SSNOII tables do account for space-based opticals as well)
4. Accuracy Percentile (assume 95^{th} percentile accuracy)
5. Accuracy at specified time from Two-Line Element Set (TLE) epoch (assume 18 hours from epoch)

6. Number of tracks per Length of Update Interval (LUPI), which is based off orbital regime and EDR group

Using information given in the TLE, and some assumptions, inputs 1-5 can be determined and will produce a curve that shows desired positional accuracy versus how many tracks (over the Zulu Day) are needed on that object to accomplish the corresponding accuracy. An arbitrary example of this curve is shown in Figure 1.

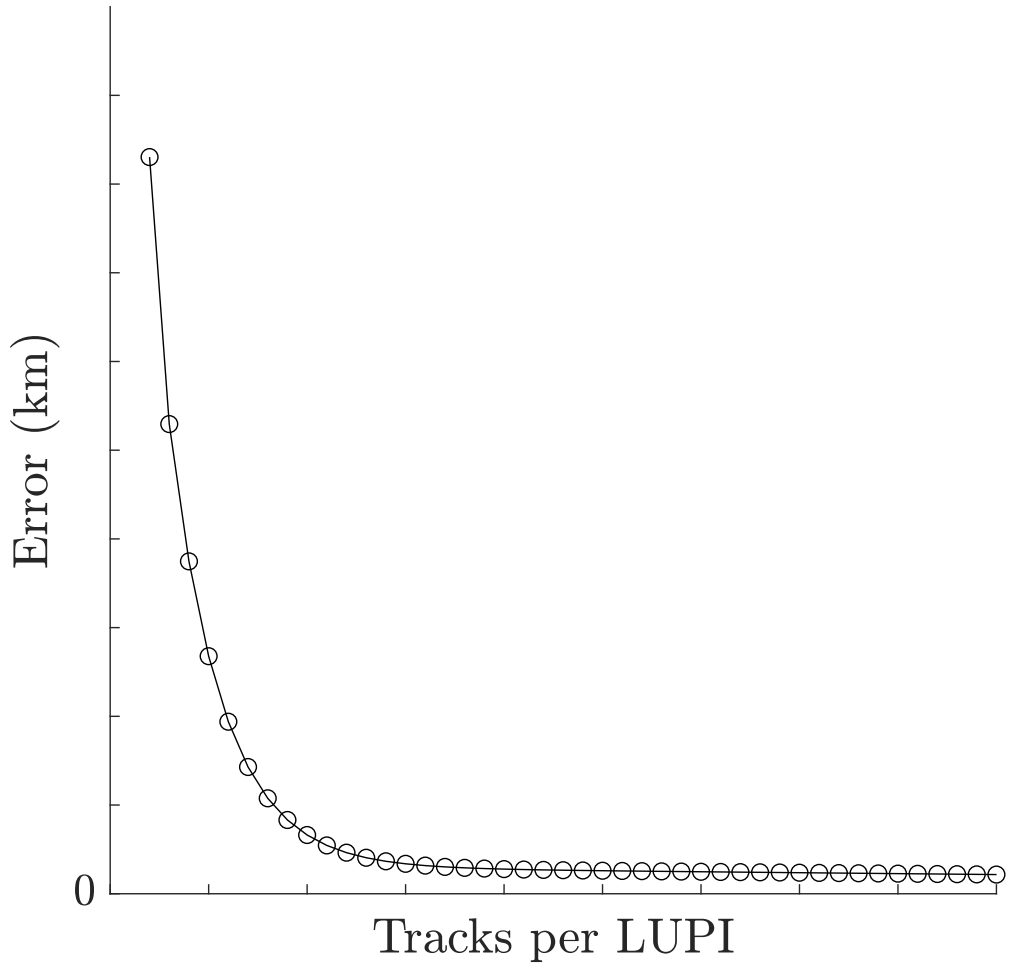

Figure 1. An arbitrary SSNOII curve.

In this simulation, positional error can be received from these curves for each object which can help us arrive at a final metric[13].

Let

$$TPE_T = \sum_k TPE_k(T_f)$$

where $TPE_k(T_f)$ is the error obtained (in kilometers) when SSNOII is updated with ψ_k's track count at time T_f. This metric can be further broken down to look at any set of RSOs (by orbital regime, EDR group, etc).

5 Setting up a Test Environment

For simulation, MATLAB version R2018a was used on a Dell Latitude 7480 running Windows 10 64-bit, with an Intel®Core™i7-7600U Central Processing Unit with a 2.80 gigahertz processor and 16 gigabytes of installed Random Access Memory.

5.1 Parameter Determination and Problem Assumptions

- To maintain the concept of "tactical" in an SDA sense, $t = 192$ is chosen (which corresponds to a timestep every seven and a half minutes).
- Simulation time, $T_f = 7 * t$, provides a continuous week of data for testing.
- Number of sensors $n = 23$, where each "sensor" more accurately describes a "sensing component" that belongs to a sensor site—all sensors are modeled from real sensors in the SSN, and include mostly radars and some ground-based opticals.
- Multiple sensor capacity assumptions will be explored via multiple tests, as shown in Table 1. Let $c_i = q * \max_T \sum_k v_{i,k}$, $q \leq 1$, to which each sensor's capacity in the simulation is some ratio q of the total things that can be seen by σ_i.

Table 1. Capacity test values

C	$c_i = 1$	$q = 0.003$	$q = 0.006$	$q = 0.01$	$q = 0.02$	$q = 0.03$	$q = 0.04$
D	4416	10560	20544	30144	62208	93504	126720
C	$q = 0.05$	$q = 0.06$	$q = 0.07$	$q = 0.08$	$q = 0.09$	$q = 0.10$	$q = 0.12$
D	158016	190464	222144	254400	285888	318336	382272
C	$q = 0.14$	$q = 0.16$	$q = 0.18$	$q = 0.20$	$q = 0.35$	$q = 0.50$	$q = 1.00$
D	445824	509952	574464	638784	1117632	1598784	3199488

- The objects used in testing are derived from a current catalog, with a distribution of LEOs, MEOs, GEOs, and HEOs
- Number of RSOs, $m = 17568$, or 13667 LEOs, 2612 MEOs, 950 GEOs, and 339 HEOs
  - In order to explore the utility function more thoroughly, two "active" RSOs will be chosen. These RSOs will have a relatively short revisit expectation, and have the ability to be chosen at many timesteps (i.e. their visibility matrices are dense). A particular object in LEO, $\psi_\lambda = \psi_4$ will be chosen, and a particular object in GEO, $\psi_\gamma = \psi_{476}$ will be chosen. Keep in mind that with the number of RSOs in the 17000s, both RSOs have interest levels within the top three percent.
- The current SP Tasker paradigm does not use a unique, ordered priority list and instead uses "binning" in which objects are placed into five separate bins where the effective priority level of any m objects are $\{1, 2, 3, 4, 5\}$ (1 being "most important"). Part of the development of this new collection paradigm is to give worth to *each* object in the catalog and be able to prioritize m objects into a list of $\{1, \ldots, m\}$ priority levels. p_k is defined as

$$p_k = k, k \in \{1, \ldots, m\}$$

  - To correspond with real objects, with otherwise real qualities, each Satellite Identification (SID)—converted from the list of m TLEs—maps to a p_k. For simplicity sake, and to avoid data sensitivity, the ordered list of SIDs map directly to an ordered list of p_k. Hence, the older the object is in the catalog, the more importance it has in the simulation (though this does not affect the simulation in any way other than determining a tie in the utility calculation).

- The r counter is initialized to "1" for each object at $T_0 = 0$. The $\hat{r}$ parameter is binned to represent potential similarities in revisit expectations used today. Let

$$\hat{r}_k = f_1(p_k) \in \{1, \ldots, 5, \frac{t}{32}, \frac{t}{24}, \frac{t}{16}, \frac{t}{8}, \frac{t}{4}, \frac{t}{2}, t\}$$

  The first five objects have a distinct revisit expectation, while every object afterward is binned. In other words, the first five objects arbitrarily become the most "important" objects with the highest revisit rates (required to be tracked every 1,2,3,4 or 5 timesteps). The following objects get placed into bins of ascending requirements that require 32 tracks per time measure t, then 24 tracks per time measure t, then 16 tracks, 8 tracks, 4 tracks, 2 tracks, and eventually 1 track per time measure t. (The distributed revisit expectation is shown in Section 11, Figure 22.)

- The o counter is initialized to "0" for each object at $T_0 = 0$. Let $\hat{o} = 3$ for all LEOs, $\hat{o} = 2$ for all MEOs and GEOs, and $\hat{o} = 1$ for all HEOs. Notice how, in general, the revisit expectations are stricter than that of the observation expectations. For some MEOs, GEOs and HEOs, $\hat{r} = \hat{o}$, and although some of the observation expectations are stricter than the revisit expectations of the same objects, the majority of RSOs have a stricter revisit expectation.

- Using Architecture Trades and Sensor Assessment Tool (ATSAT), the list of TLEs is propagated for n real sensors in the SSN (ATSAT is a MITRE developed tool used primarily for STM, SSN validation, missile defense analysis, and satellite coverage analysis that has contributed to major SSA architecture decisions of the past decade). This produces a binary matrix, $V \in \mathbb{B}^{n \times m \times T_f}$, to which $v_{i,k}(j)$ can be derived from at the correct timestep, T, in the matrix V. An illustration that demonstrates the number of timesteps that each object is visible is shown in Figure 2. Note that there are some objects that can be seen at every timestep by at least one sensor, yet most of the objects are only visible around half of the total timesteps. Notice how LEOs are generally seen less than the other objects. This is because LEOs orbit the earth on an average of about 90 minutes. Though they are typically visible by a sensor multiple times throughout the day, they aren't typically visible for very long consecutively. This contrasts the nature of a GEO object which doesn't move very much relative to a sensor that might be tracking it, and it is therefore potentially visible by one sensor the entire day (sensor physics allowing). A distribution illustrating the total count of times each ψ_k appears in V is shown in Figure 3.

- Noise is added to the simulation in two ways

  1. Expect 5 percent or less of planned collections to become "missed," in which case those items are added to the bounty list until they become a confirmed collect at a later timestep. These objects are taken away from the schedule, and hence can negatively effect the revisit, priority, observation and diversity metrics.

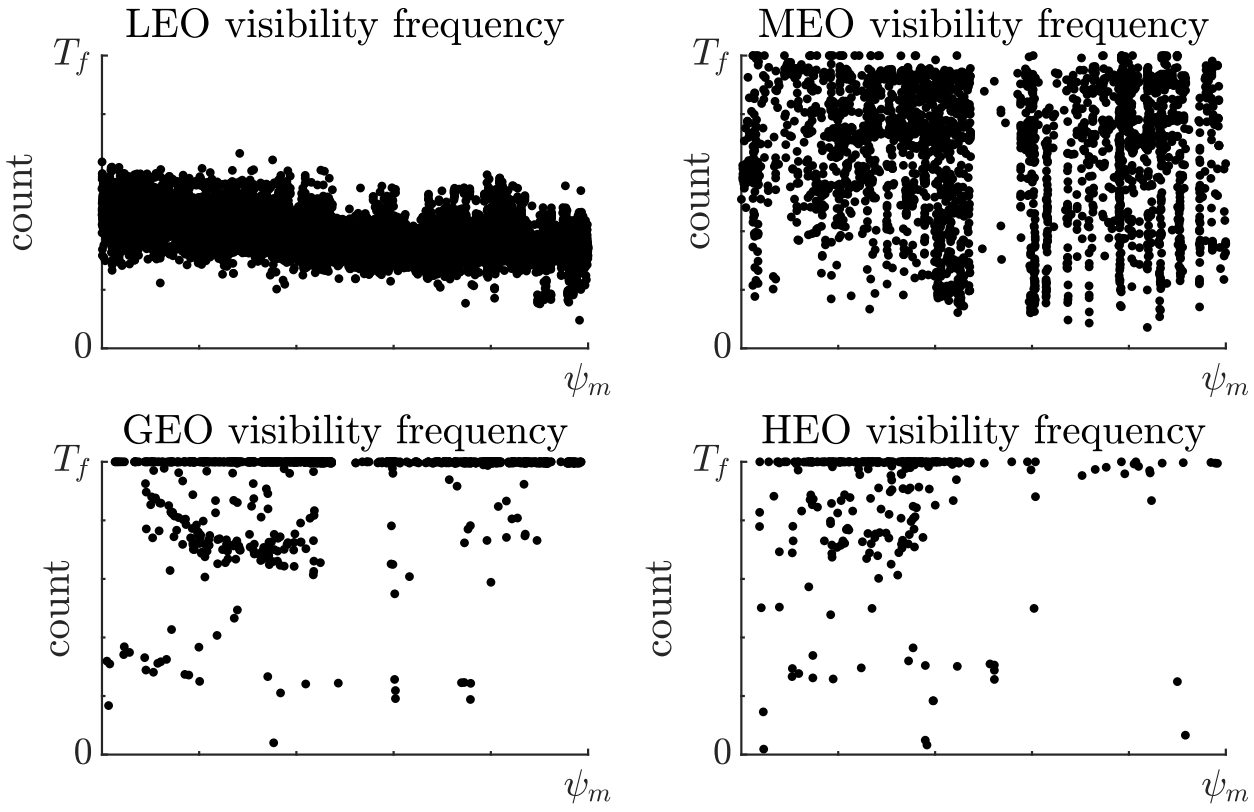

Figure 2. Distribution of how many timesteps each ψ_k is visible in the simulation

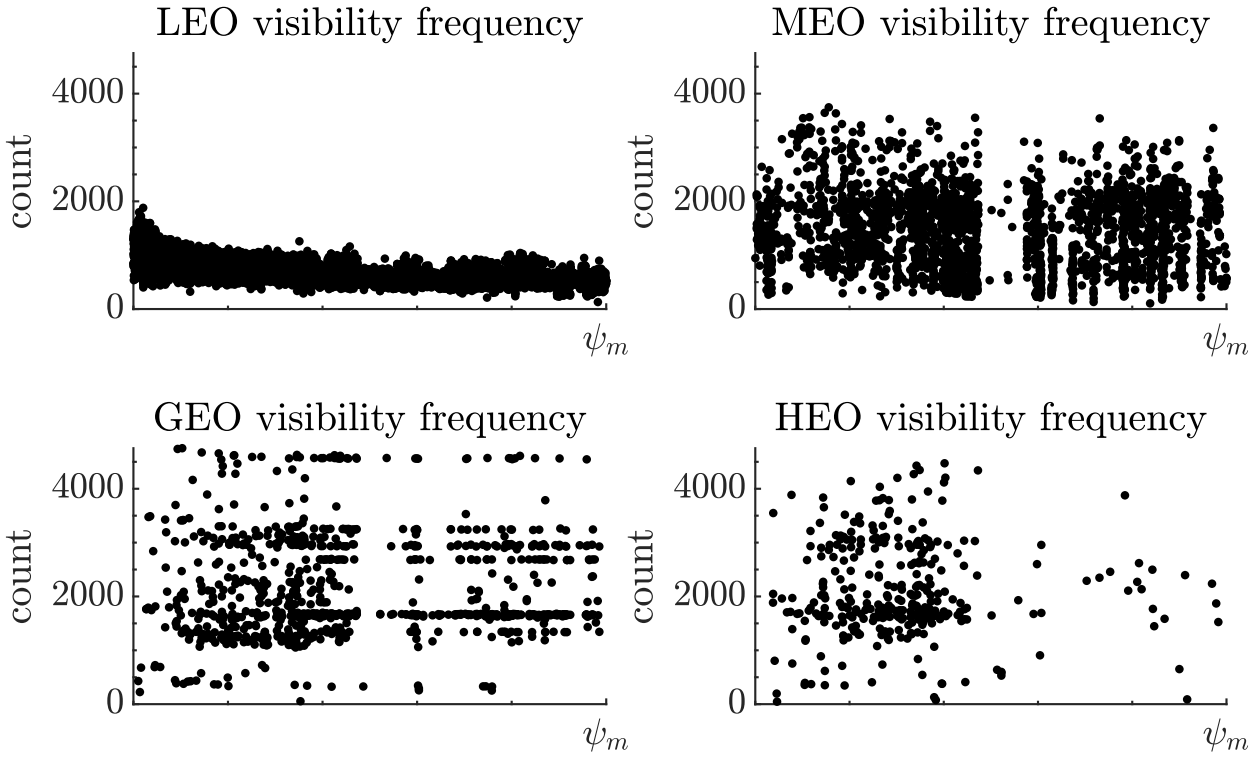

Figure 3. Distribution of how many times each ψ_k is visible in the simulation

2. Expect 5 percent (but not to exceed network capacity) or less of what is visible at any point in time to be the number of serendipitous collects to be added to the collection scheme. These collects are not put into the schedule as they are accidental collects, and hence do not contribute to the multiview scoring metric nor the diversity metric. However, these collects are relevant to r, o values, so those are updated appropriately which can positively affect the revisit and observation metrics. Denote the set of serendipitous collects at time j as $\mathcal{H}_j$ for reference.

The noise helps add some element of reality to the simulation while testing the scheme, although the percentages chosen are likely unrealistic. In addition, noise was exaggerated slightly for $\psi_\lambda, \psi_\gamma$ to demonstrate qualities of the system. **This is the only part of the simulation that is stochastic, and it does affect the outcome of the results marginally**. An in-depth study is not included that quantifies the effects of the noise, but the noise is necessary to demonstrate some of the reactive properties of the value function.

5.2 Picking a Utility Function

Recall the formulation for utility in the LVF:

$$U_{i,k}(j) = v_{i,k}(j) * f(r_k(j); \hat{r}_k) * g(o_k(j); \hat{o}_k) * h(p_k)$$

As formulated earlier in Section 3.2, there are a few principle qualities that must be upheld when choosing the utility function. A summary of those qualities is here:

- It must be possible for what we normally consider to be a Priority 3 or even Priority 5 objected is of higher interest at a specific time or place than a Priority 2 object. Note there are situational cases tied to specific scenarios and time or place in orbit can be considered as well.
- Value increases as time between tracks increase.
- There is a difference in the rate of value change between an object that is coming up to its revisit expectation and when it surpasses its revisit expectation.
- An object that hasn't been seen is usually more valuable than an additional track on the same object.
- All valid collections provide utility to the network.

These qualifications on the utility function provide soft groundwork for some general qualities in choosing the particular function that governs value of collect.

- $f \in [a, \infty), \forall (r_k, \hat{r}_k), a > 0$
- $f' > 0, \forall (r_k, \hat{r}_k)$
- $f''(\hat{r}_k) = 0$

- $g \in (b, B], \forall(o_k, \hat{o}_k)$, where g is bounded by some constant $g(0) = B$
- $g' < 0, \forall(o_k, \hat{o}_k)$, which implies $\lim_{o_k \to \infty} g = b$
- $h' < 0, \forall p_k$
- Since $p_k = k$ and h is time invariant, it follows that $h \in [\min_k h, \max_k h]$, so $h \in [h(m), h(1)]$

Let

$$f(r_k(j); \hat{r}_k) = 1 + \frac{r_k}{\hat{r}_k} * (\tanh(r_k - \hat{r}_k) + 1), \qquad f \in (1, \infty)$$

$$g(o_k(j); \hat{o}_k) = 1 + \frac{2 * \hat{o}_k}{2 * o_k + 1} * (\tanh(\hat{o}_k - o_k) + 1), \qquad g \in (1, 13)$$

$$h(p_k) = \frac{p_k + 1}{p_k} \qquad h \in (1, 2]$$

with the limits of g dependent on parameter selection as defined in Section 5.1. A general look at the algebra associated with this particular utility function can be found in section 9.

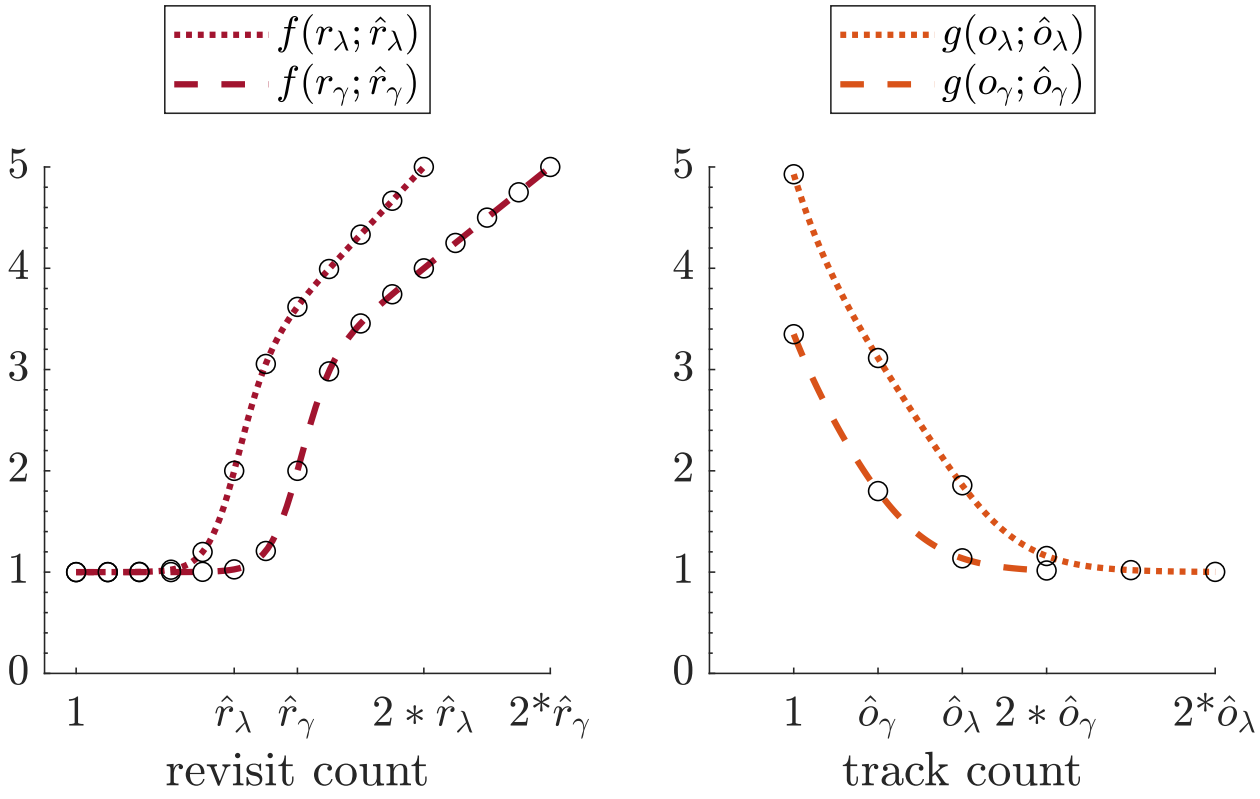

Figure 4. f, g **components of the utility function for** $\psi_\lambda, \psi_\gamma$

Components on f, g can be examined in Figure 4. Note the range of each utility component, given the parameter assumptions in Section 5.1. In the LVF, since elements of U are sometimes raised to an exponent value greater than one, then all elements of U that are positive but less than one

might be punished. To avoid a potential decrease of value in the LVF logic, note that a small scalar additive is included in the f, g components of the utility function. The utility function is then

$$\begin{aligned} U_{i,k} =& v_{i,k} * \frac{p_k + 1}{p_k} * \dots \\ & \left(1 + \frac{r_k}{\hat{r}_k} * (\tanh(r_k - \hat{r}_k) + 1)\right) * \dots \\ & \left(1 + \frac{2 * \hat{o}_k}{2 * o_k + 1} * (\tanh(\hat{o}_k - o_k) + 1)\right) \end{aligned}$$

where $U_k \in (1, \infty), \forall k | v_{i,k} = 1$.

In these simulations, $\alpha = \frac{11}{10}$ was chosen for the consideration of bounty-list objects in the LVF (as referenced in Section 3.2). This value is large enough to have a noticeable effect on the value of collect but small enough as to not exaggerate the results in the following simulations.

5.3 Picking a Global Value Function

All the metrics in Section 4 already exist as metrics in $[0, 1]$. Assuming each metric has equal weight in the determination in the score of a collection scheme, a valid and easy choice for the GVF is

$$GV_T = \widetilde{dv_T} * \widetilde{mv_T} * \widetilde{rv_T} * \widetilde{pv_T} * \widetilde{ov_T} \in [0, 1]$$

which of course makes the "target" score a perfect one out of one. Because this is a score that is based on counting what a scheme *did not* do correctly, this allows room for multiple schemes to reach the same score (i.e. a perfect collection scheme may not be unique).

5.4 Outcome Goals

As alluded to throughout the paper, the main objective is to

- Obtain collection scheme via proper selection of methodology and LVF.
- Demonstrate that this may occur with minimal violations at some capacity using metrics from GVF.

To indicate this may have occurred, there are a few key outcomes that are desired, listed as follows:

1. $\exists C : \widetilde{dv} \approx 1$

   By construction, every object that was selected to use in the simulation is visible at least once by a sensor. There should generally be more than enough opportunity to appropriately see every object once a day, and some of these objects (particularly GEOs) can be seen at nearly every timestep in the simulation. Achieving this point at some capacity assumption strengthens the idea of good catalog keeping as well as persistence.

2. $\max C \rightarrow \min rv$

   This is a valuable "proof of concept" token. It is unrealistic to assume that the arbitrary revisit expectations set forth can properly encapsulate the physical nature of all objects in the catalog. In other words, even if it is assumed that everything that is visible is placed in the collection scheme, and there is no noise to prevent those collects from occurring, it is unlikely all objects will meet their revisit time. However, until that capacity limit is reached to provide an actual bound on how low the revisit violation metric can be, it is hopeful that revisit violations decrease with increase of resource assumptions.

3. $\lim_{C\rightarrow\infty} pv = 0$, and $\widetilde{pv}_T = 1$

   The priority score is good at comparing multiple schemes to each other, but in particular it is important to show that if there is enough sensing capacity, then the important objects will always be tracked ahead of lesser ones. Even though the priority metric should take a similar shape as the revisit violation metric, the magnitude of the priority metric cannot be reliably determined from the revisit violation metric. The key element to demonstrate here is that if every sensor was able to grab everything in its field of view, then there would be no priority violations possible to occur, regardless of revisit expectation. This is important beyond the scope of the presented methodologies and LVF.

4. $\exists C : \widetilde{ov}_T \approx 1$

   Even if only poor global score is ever achieved, it is necessary to be able to show that some arbitrary level of accuracy requirement can be met, and real sensor models for sensor-object visibility can lead to some assumed "catalog accuracy." For these simulations, a score close to perfect needs to be achieved for some capacity assumption, even under loose accuracy assumptions. In other words, if the network has infinite capacity and objects aren't being seen enough then it is the fault of the network (or the requirements) and not the collection scheme.

5. $k \in \mathcal{S}_j \wedge \mathcal{S}_{j+\hat{r}_k}$

   This metric is what sets the defined methodology apart from the rest. If an object appears in the schedule at time j, the goal is to achieve persistence by getting it back in the schedule near the time that occurs when the revisit expectation occurs. It isn't just sufficient to minimize the revisit count, but in particular to try and force the times that objects are seen to a particular distance apart (in this case by a distance of $\hat{r}_k$). If this statement doesn't hold, then there is a large variance of revisit counts for an object which means that accuracy goes down and persistent awareness is reduced.

6 Standalone Test Results

The only parameter that was not specified for testing was capacity. Because track capacity within this testing environment is ill-defined, an assumption on SSN capacity is almost surely to be wrong for some or even most cases. Empirical data exists on the SSN that describes sensor track production from site to site but that is effected by the current tasking methodology, which is likely sub optimal. Hence, that data might serve a basis for a capacity assumption but can't really be used to form any type of concrete assertion. To that end, what will be demonstrated through testing is the effect of capacity on the system as a whole. In practice, the idea is to task the sensors until they indicate they are full (which will be briefly described in the future work section), and cannot be done in this stage of the model design when there is no actual "communication" between the sensors being simulated.

The LVF prevents objects from being tasked by multiple sensors. In addition to this, collects in $C(j)$ that are also in $\mathcal{S}_j$ are treated as a single collect. Because of this, the multiview score remains a perfect "1" using the defined methods. This metric will be more helpful later (as described in the future work section) when comparing different schemes to each other, where other methodology might allow room for duplication of work. In practice, there are certainly times when it might be necessary to task multiple sensors to look at a particular RSO at the exact same time, but that is also outside the scope of the current problem and hence will not be considered in the following simulations.

6.1 Extreme Resource-constrained Environment

Let $c_i = 1$, i.e. every seven and a half minutes each sensor can have but a single task.

6.1.1 Results for $\psi_\lambda, \psi_\gamma$

After seven days of simulation time, Figure 5 shows various qualities about $\psi_\lambda, \psi_\gamma$. Note that the axes between the left and right subplots are measured in different units, but are the same numerically (i.e. $2 * \hat{o}_\lambda = 3 * \hat{o}_\gamma$).

It is clear that this is a resource-lean environment when two important objects, $\psi_\lambda, \psi_\gamma$—which have relatively dense visibility matrices— are reaching TVT of 20 times (or more) their expected values in the simulation (top subplots). This is causing a relatively large quantity of revisit violations and priority violations by both objects, especially the object in GEO. By comparison, the observation expectations are still, in part, met for ψ_λ even though ψ_γ cannot meet its expectation at all during the week-long experiment (illustrated in the middle subplots). This is likely not the case for less-important objects, but at least some of the most interesting objects can get close to their required positional accuracy when under extreme resource-load constraints. The results imply that there will still be some observation violations caused by this.

Both objects, $\psi_\lambda, \psi_\gamma$ are put onto the bounty list during this trial. ψ_λ is actually put onto the bounty list on two separate occasions, one of which stays on the bounty list for multiple iterations (including a point at which ψ_λ cannot be seen by any sensors in the network). The utility of collect

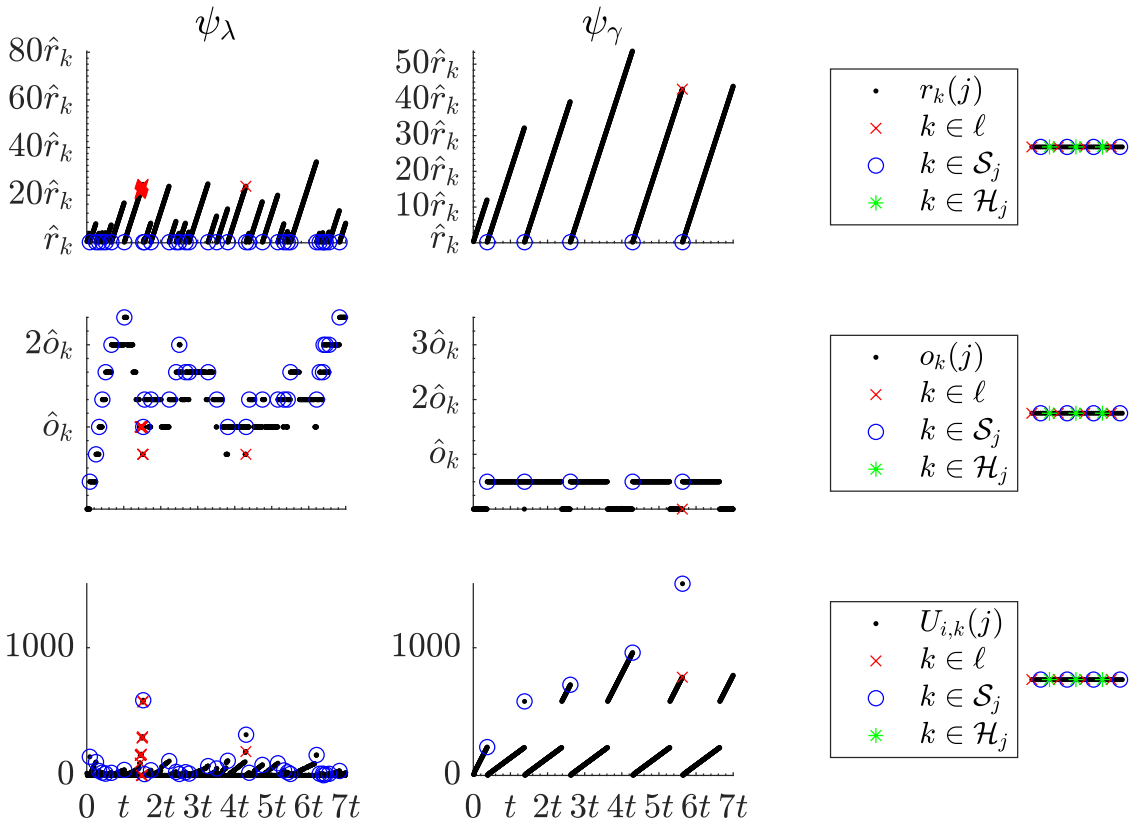

Figure 5. f, g **components and** U **over simulation time under extreme resource limitations**

on each object clearly climbs when on the bounty list (shown in the bottom subplots), and when captured decreases again significantly in value.

Important to note in this particular example is that ψ_λ is kept at a generally lower utility of collect than ψ_γ. In fact, this is true even though ψ_γ is visible by a sensor at literally every single timestep in the simulation, when ψ_λ is not visible for many timesteps it is in the simulation. The natural assumption would be that since the object in GEO is always visible, perhaps it would maintain a generally lower utility of collect because it is always available to be seen when the value rises to a certain point. Meanwhile, an object in LEO might not follow this trend because the moment it becomes valuable, if it is in a particular orbit around the Earth such that no sensor can see it for another 90 minutes, the value of collect may suddenly skyrocket on occasion and hence maintain a generally larger utility of collect value. As demonstrated by this test, this is not necessarily true. In this case, ψ_λ and ψ_γ are not generally visible by the same sensors. Hence, there may never be a point in which ψ_λ is favored by a sensor in value over ψ_γ. In addition, ψ_γ is competing with other objects that are rarely visible, or even more important GEOs (regardless of how often they are visible) on the sensor or two that can commonly see it.

6.1.2 Global Results

After seven days of simulation time, Figure 6 shows various qualities about the simulated collection scheme for the SSN.

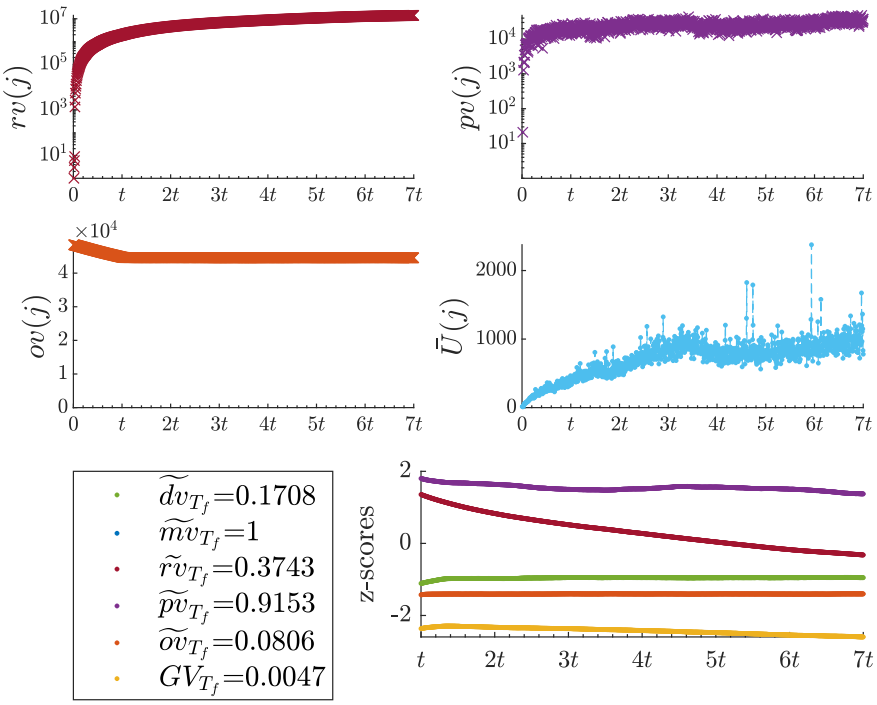

Figure 6. Metrics under extreme resource limitations over simulation time

In the first subplot (top left corner) is a log-scale plot of the count of revisit violations over time. It is no surprise that in an extreme resource limitation there is a linear increase of revisit count over time. Without enough resources to satisfy the expectations set, the system is only able to handle so many RSOs, and that is immediately apparent. The log-scale plot of the priority violation count (second subplot in the top right) reflects this as well. Priority violations soon hover around a hundredth of the priority violation count in the simulation. Though the revisit violations continue to increase throughout time, the priority violations are cyclic and do not grow in severity over time like the revisit count does. Upon first look, any quantity of priority violations seems bad, where "interesting" objects are being bumped for less interesting ones. In an extreme environment, this is the case for multiple reasons. The third subplot (middle left) shows that the observation requirements aren't being met as a whole. This metric implies that catalog accuracy will go down over time because there are simply not enough resources available to achieve accuracy expectations. From what is demonstrated graphically about the revisit and observation counts, it can be determined that the value of collect on these objects must also be growing in time. Whereas g in the utility function is bounded, f is not. So while g provides a starting point for the value on objects not being seen enough times for accuracy purposes, f is making that quantity grow and grow over time. It is unlikely in this environment that this trend would ever find equilibrium, and this is demonstrated in the fourth subplot (middle right). The average value of collect on all visible objects, denoted as $\bar{U}$, is shown to climb over time. The average value of collect plot shows some of the immediate effects of the noise. It can be seen that there are periods in the average value of collect that spike suddenly on a particular timestep. It is likely because a lot of potentially lower-value collects were taken off the schedule (treated as missed collections and hence put on the bounty list) which spiked each individual's value of collect.

The final graphic on the bottom shows all the metrics based off the score (recall that this is determined

using the problem dimension size and includes capacity in some of the metrics). Instead of showing the actual score over time in the final graphic, the z-score of each score is shown instead, to help more easily distinguish the scores. No statistical implications are trying to be made but since all the metrics including GV are $[0, 1]$, the z-score can be a helpful mapping to show relative differences between scores when very close to a perfect score (recall that the z-score computes the number of standard deviations away from the mean that a value between 0 and 1 lays—here a score of 0.50 would equate to a z-score of 0 and a score of 0.997 would equate to a z-score of 3). Since $\max_k \hat{r} = t$, the first day of simulation was excluded in hopes of illustrating more representative scores.

As the assumptions on the LVF imply, the multiview score stays perfect throughout the simulation (and therefore does not have a numeric z-score). The priority violation score is still relatively high given the problem size, but still is not perfect. Meanwhile, the observation and diversity score reached an equilibrium around the first day mark in the simulation, implying that only about 8 percent of all objects meet their accuracy requirements, and only about 17 percent of the catalog gets scheduled in a day, on average. The revisit score gets worse and worse over time, and though it never reaches the worst score possible, there is a negative trend if the simulation time is longer. This, of course, also causes the global value score to decrease over time, which is already very low due to the observation score.

6.2 Resource-limited Environment

In this subsection, let $q = .04$. In this case, $C * t = 126720$ possible collects allocated to the SSN per day.

6.2.1 Results for $\psi_\lambda, \psi_\gamma$

After seven days of simulation time, Figure 7 shows various qualities about ψ_k. Note that the axes between the left and right subplots are measured in different units, but are the same numerically (i.e. $10 * \hat{o}_\lambda = 15 * \hat{o}_\gamma$).

The TVT has now been significantly lowered under more relaxed resource constraints. For ψ_λ, TVT is now about ten times less in severity on average. ψ_γ made even better improvements, dropping to about within twice its revisit expectation, which is an improvement of closer to 20 times better than the previous test. It is clear that both objects have dramatically increased their track rate, and now both objects meet their observation requirements nearly 15 times over. It is interesting to note that both objects, the one in LEO and then one in GEO seem to be violating/exceeding their expectations by the same magnitude. It is also interesting to see, in particular with ψ_γ that the bounty effect doesn't seem to have a drastic effect on when it is recaptured. In general, it almost looks as if the object would have been captured at the same point regardless of whether or not it was on the bounty list—a potentially good side effect to the idea of "persistent" tracking.

The value of collect between the two RSOs has become a lot closer. This could be a side effect of a more stable collection scheme. In theory, if every object in the catalog was to have the same value of collect, then all objects would be similarly close to violating their revisit or observation expectations, and hence there would be some equilibrium in the system. Of course, there still is

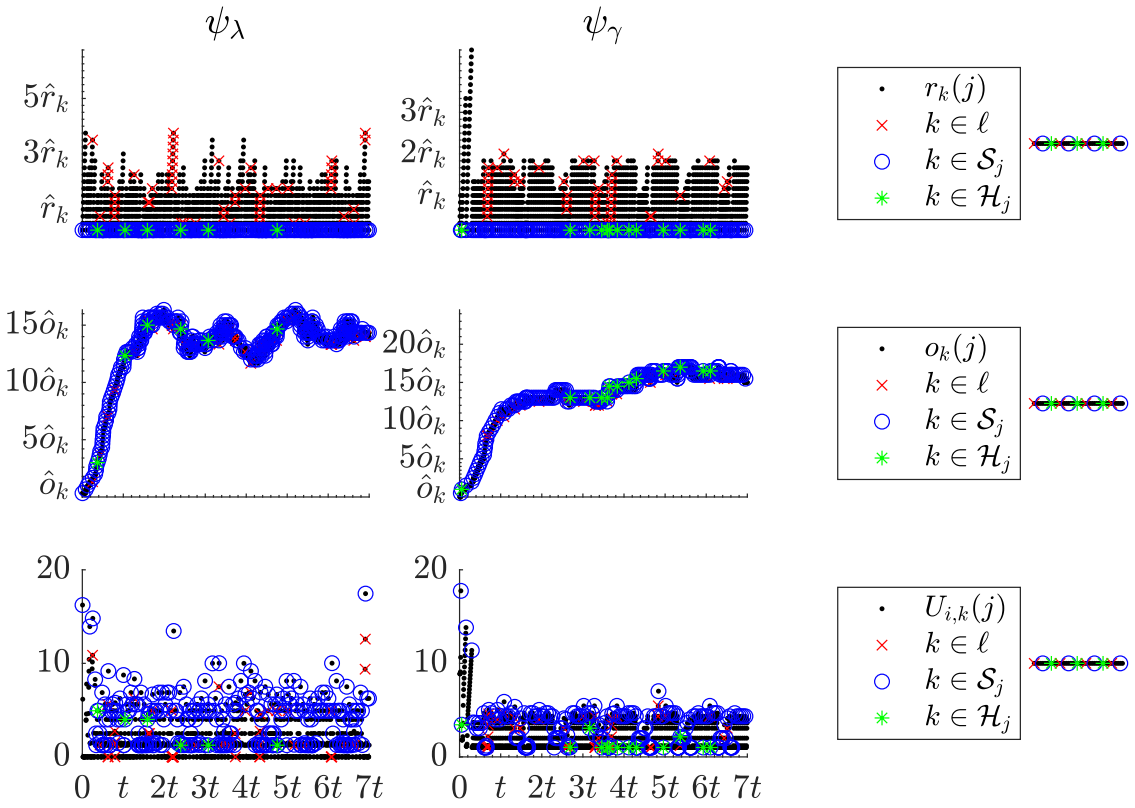

Figure 7. f, g **components and** U **over simulation time under resource limited constraint**

some variation between value of collect between the two objects (and these are only two objects out of over 17,000). It is curious to see that both objects are sometimes being collected with $U_k \approx 1$. This might be a side effect of picking two objects that are more commonly visible than others, where there are then periods when a particular sensor is only able to see a handful of objects, all of which can fit within the duty cycle allocation of the sensor. Hence, it is good to see that this is occurring, because it lends itself nicely to the idea that sensor's resources can be utilized even when the objects it has to look at aren't necessarily in dire need of being seen at that moment. This also supports freedom in the system with more realistic noise. If an object starts moving around and one of these sensors can be utilized to recover that object, the regrets the sensor has on the objects it chooses not to observe have low impact on the system due to dropping objects from its schedule that have low current value of collect.

6.2.2 Global Results

After seven days of simulation time, Figure 8 shows various qualities about the simulated collection scheme for the SSN.

A big change that occurs globally with relaxed resource constraints is the delinerization of the revisit count. There is apparent periodicity that arises as well, though interestingly the peaks that occur toward the beginning of the simulation begin to calm before the simulation is over. This could mean that there is some stabilization in the system leading to an equilibrium over time. This is somewhat evident in the priority score count as well. Peaks in the priority violation count occur

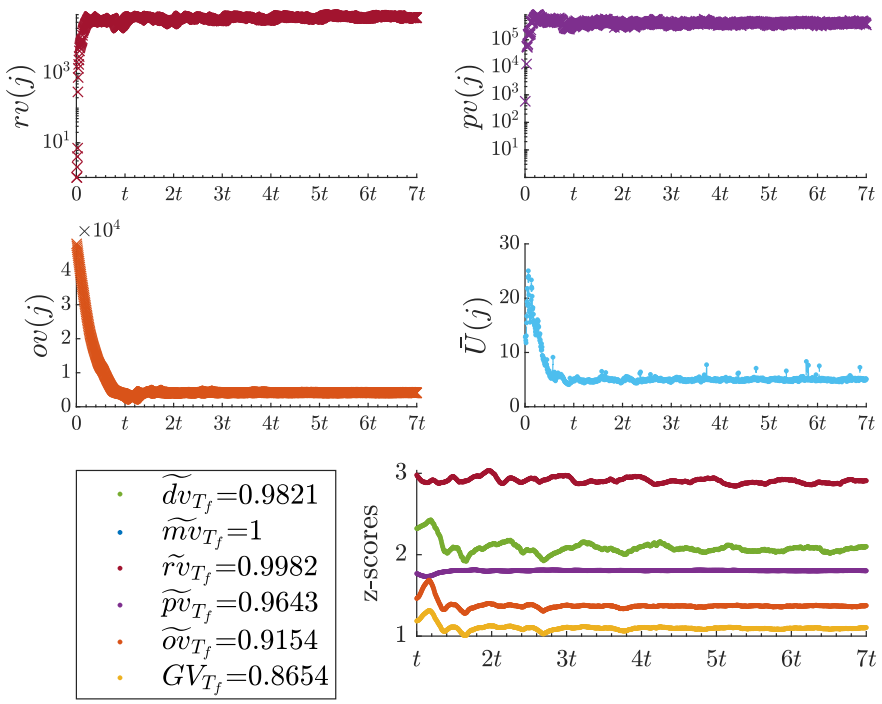

Figure 8. Metrics under limited resource constraint over simulation time

periodically in correspondence with the revisit violations, and slowly die out over the simulation time. It is important to notice that less revisit violations occurred in this simulation with increased resources, yet the priority violations increased. Noise certainly can contribute to this, but it is likely that the main reason is actually due to increased capacity–now that some sensors have access to multiple collects in a single timestep, more objects are able to be placed in the schedule and hence more instances of violations are able to be counted. Referring to the score chart, though the actual violation count is higher, the relative score at the end of the simulation is increased.

The track count reaches equilibrium more quickly than the more resource limited simulation. Instead of reaching a track capacity at the end of the first day, there are enough sensing resources to satisfy the observation requirements of over 90 percent of the catalog before the first half of the first simulation day is over, and the observation expectations are generally able to be maintain that accuracy in the simulation, however, noise from the bounty list might be part of what is keeping all objects from reaching their expectation.

The average utility of each RSO is more distinguishable as the system fights to achieve an equilibrium. Notice how the fluctuations in the utility metric also follow the trend in the revisit count. To see that the average utility isn't drastically decreasing over time suggests that the value of the objects remains high. In turn, this suggests that an increase in resources might be necessary to meet the revisit expectations that are driving the utility of collect up. The system's ultimate goal is to have the maximum utility of each object be close to the minimum expected utility value, which means that violations are less likely to occur due to largely available resources.

The global value is largely improved in this simulation. Most metrics hover around a perfect score, but the diversity metric seems to be the global value component that is preventing a higher overall global score. It looks like the global score is actually highest in the first day, but generally decreases

afterward. More resources might be the key to solve this to achieve a more consistent, and higher, global value score.

6.3 Resource-Abundant Environment

In this subsection, let $q = 0.20$. In this case, $C * t = 638784$ possible collects are allocated to the SSN per day.

6.3.1 Results for $\psi_\lambda, \psi_\gamma$

After seven days of simulation time, Figure 9 shows various qualities about ψ_k. Note that the axes between the left and right subplots are measured in different units, but are the same numerically (i.e. $3 * \hat{r}_\lambda = 2 * \hat{r}_\gamma$).

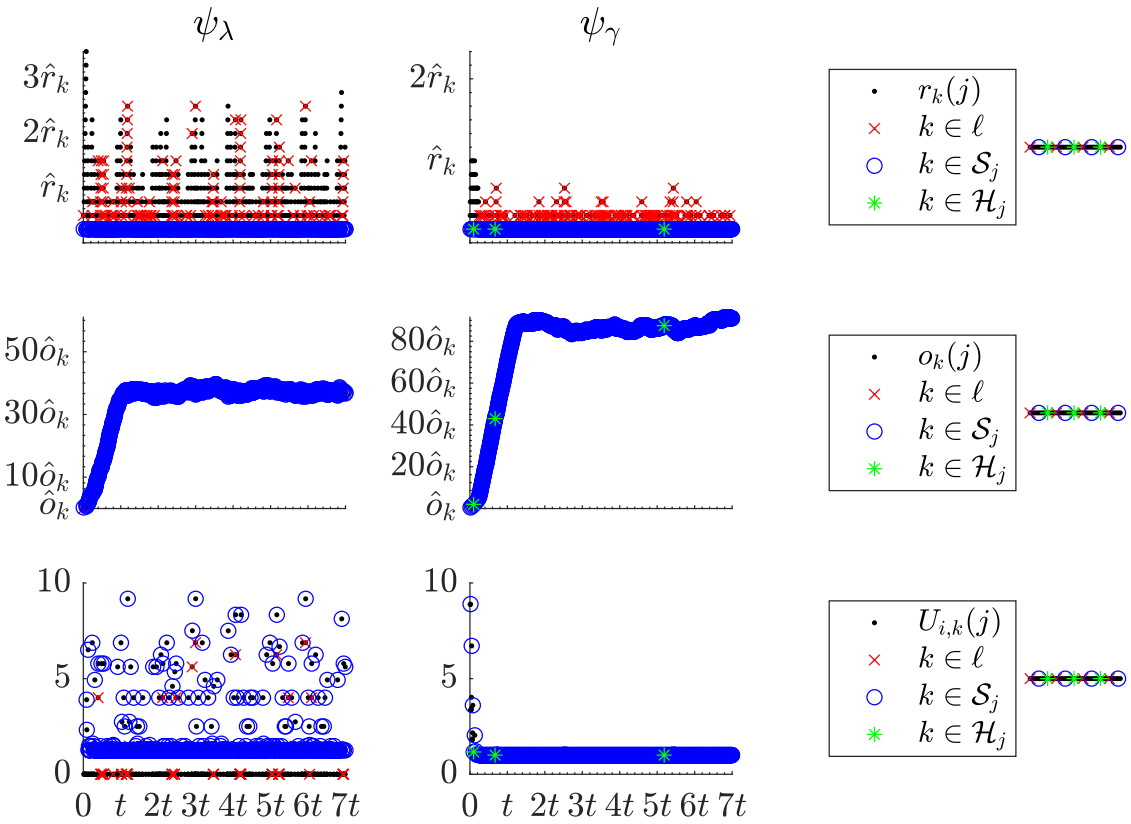

Figure 9. f, g **components and** U **over simulation time under resource abundant conditions**

The final test demonstrates some expected properties of the system. On one hand, now there are clearly enough resources to be able to see the GEO object, just about once every 7.5 or 15 minutes (which is less than half of its revisit expectation). Even when the object is put onto the bounty list, it is always shortly recovered (never more than 22.5 minutes later). This leads to a track rate of almost 100 times greater than the expectation. Hence, the GEO object maintains a value of collect very close to one throughout the simulation, under these resource conditions.

On the other hand, the LEO object still cannot quite hit its (rather strict) revisit expectation. This actually might be an artifact of the physics behind the LEO object. ψ_λ is still commonly captured within its revisit expectation, but the times that it is not could be because it simply wasn't visible in those timesteps. This is mostly evident in its utility plot, where although the value of collect upon collection is commonly a value of near one, there are sporadic instances where the value of collect jumps to 2, 4, 8 or more times the mode value of collect.

6.3.2 Global Results

After seven days of simulation time, Figure 10 shows various qualities about the simulated collection scheme for the SSN.

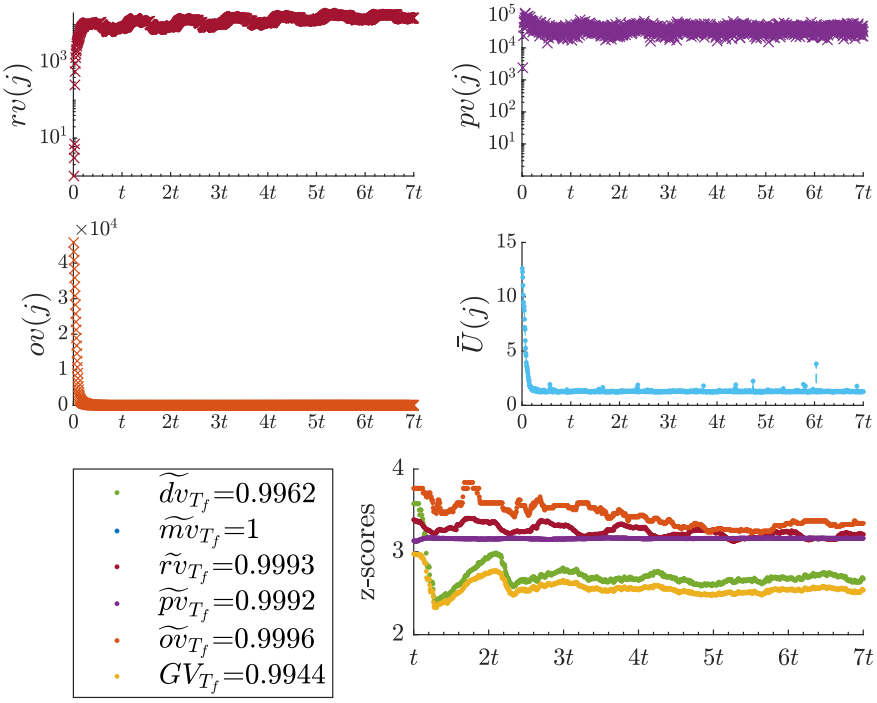

Figure 10. Metrics under resource abundant conditions over simulation time

The revisit violations may have dropped a little bit, but it is possible that the assumptions made on the strictness of revisit expectations cannot be satisfied much further, or that the noise floor due to the bounty list has been reached. This may also be the case with the priority violations. The observation violations smooth out to a near perfect score, and do so within the first portion of the day, even faster than in the previous test.

The schedule diversity is probably close to its limit, with somewhere around only 100 objects not being seen in the final simulation day. The value of collect between the objects hovers closely to a utility of one, which supports the idea that the network can handle more objects under the relaxed capacity assumptions.

6.4 Capacity Sweep

Because the other tests included artificial noise, another experiment was run without noise, to more accurately depict the effect of capacity on the collection methodology. The noiseless environment has no serendipitous collects and nothing is placed on the bounty list, so the tests are completely deterministic. Each simulation was for seven days as in the other experiments, and reflects the numbers in Table 1.

Only basic measurements were recorded, and are based on statistical means of the last day in each simulation week. Figure 11 shows the results. The first three log-log subplots show the last day average revisit violation count, priority violation count, and observation violation count respectively. The fourth log-log subplot shows the average utility score of all visible objects, over the last day for each simulation. The final semi-log subplots show the z-scores of the average diversity and global value scores of the final day of each simulation, respectively.

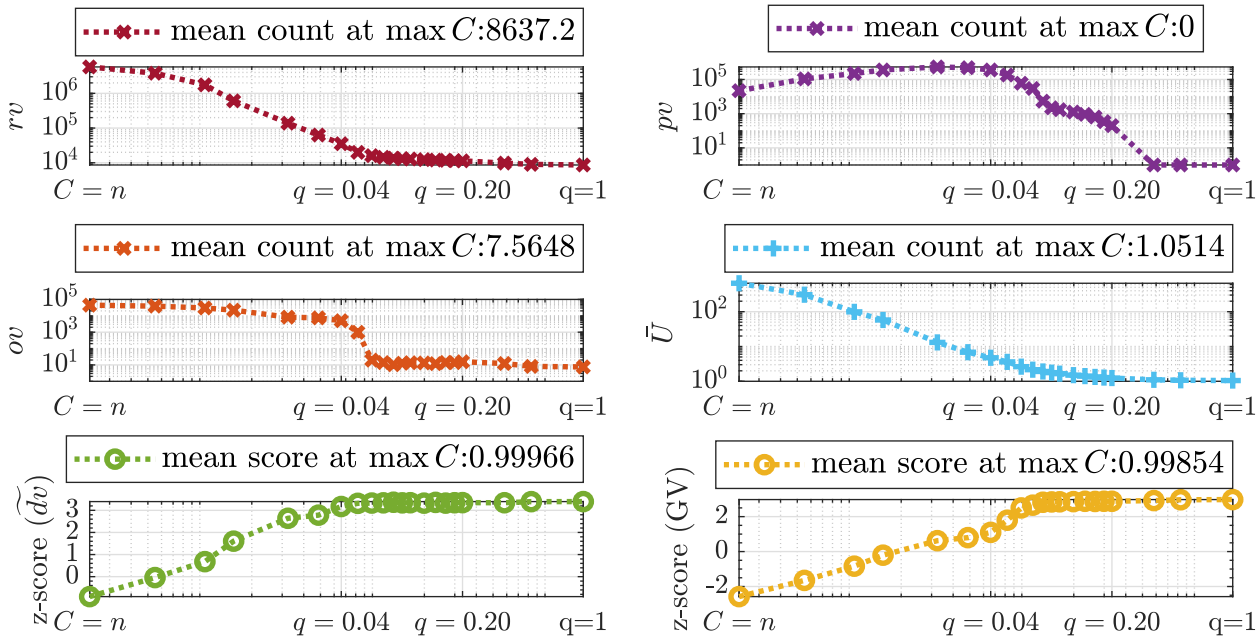

Figure 11. Noiseless capacity sweep

The revisit and observation counts demonstrate intuitive qualities as capacity increases. Revisit expectations and observation expectations are more often met when the schedule size is allowed to be larger. Of course, it is also relevant to note that even under ideal resource conditions not quite every object can meet its revisit expectations. A surprising ratio of objects end up meeting their revisit expectations, but not all. Observation expectations are nearly met, with only a small handful of objects not being able to fully meet their expectation.

The priority violations aren't as intuitive as capacity increases. It appears that at around 2 percent capacity is where priority violations occur in the greatest count, even though revisit violations are on the decline at that point. Speculations on this are discussed briefly in Section 6.2.2. By about 35 percent capacity priority violations actually completely disappear, which indicates there is never a point in the schedule such that a less interesting object is tracked instead of a more important object

if both are visible. It is interesting to note that the noise in the earlier simulation at 20 percent was two or three orders of magnitude larger than without noise.

As expected the utility of each object approaches the minimum as capacity increases because there is no longer a resource constraint. In addition, the diversity of the schedule and overall scoring of the method is highest when capacity is highest, as expected, although is begins to have diminishing returns in relative value closer to 6 or 7 percent.

6.5 Violation Time Results

This section will explore VT under noiseless conditions for each of the capacity tests run in the previous section. Figure 12 is a y-axis log box plot that shows the collection of the average revisit time for each object—normalized by its revisit expectation $\hat{r}$—over each capacity test. The green line is the "target" line, at which the revisit expectation of an object has been met. Anything above this line is in violation of the revisit expectation by a ratio of its expectation value on average. Anything below this line has a negative VT, and therefore has an average revisit time that is some proportion of its expectation. In other words, a value of 10 on this plot means that that object's revisit time is 10 times greater than its expectation, and a value of 0.1 means that the objects revisit time is generally a tenth of its expectation. It is important to note that the box plots are spaced equally on the x-axis, however the capacity values as listed in Table 1 are not linearly spaced.

The average violation time decreases monotonically as capacity increases, which is consistent with the revisit violation count findings in Section 6.4. Under the extreme resource constraint, there are no objects that meet their revisit expectation on average. In fact, this doesn't appear to begin to happen until $q \approx 0.01$. Right around $q \approx 0.04$ shows that nearly 50 percent of the RSOs start to meet their expectation values, with the median sitting right on top of the target. At a slight increase of capacity at between 5 to 8 percent collection rate of what is visible, the majority of data excluding outliers are meeting their revisit times on average. By $q \approx 0.20$ some of the outliers even start to meet the target, and by $q \approx 0.50$ there are only two or three objects that cannot meet the target on average. The limit of how good the average revisit time can be throughout the simulation is shown on the final box plot, when $q = 1$. The outliers remain similar in count to the previous two capacity tests, although the median time is much lower than the rest of the tests, where half of the objects are likely meeting their expectations t times greater than their expectation (in the case where the object's revisit expectation $\hat{r} = t$ and they are captured at every timestep—a non-interesting object in GEO for example).

6.6 Accuracy Results

The accuracy results are somewhat implied by the results with the observation scores. However, a look at the physical quantities generated by SSNOII tables provides more tangible support about the accuracy of the catalog. Figure 13 is a heatmap of the relative magnitude of error formulated using SSNOII, where yellow indicates the highest total positional error and blue represents the lowest error. The top subplot shows the relative error over time of the LEO object, ψ_λ. The mapping also shows each of the other tests, using different capacities, along the y-axis.

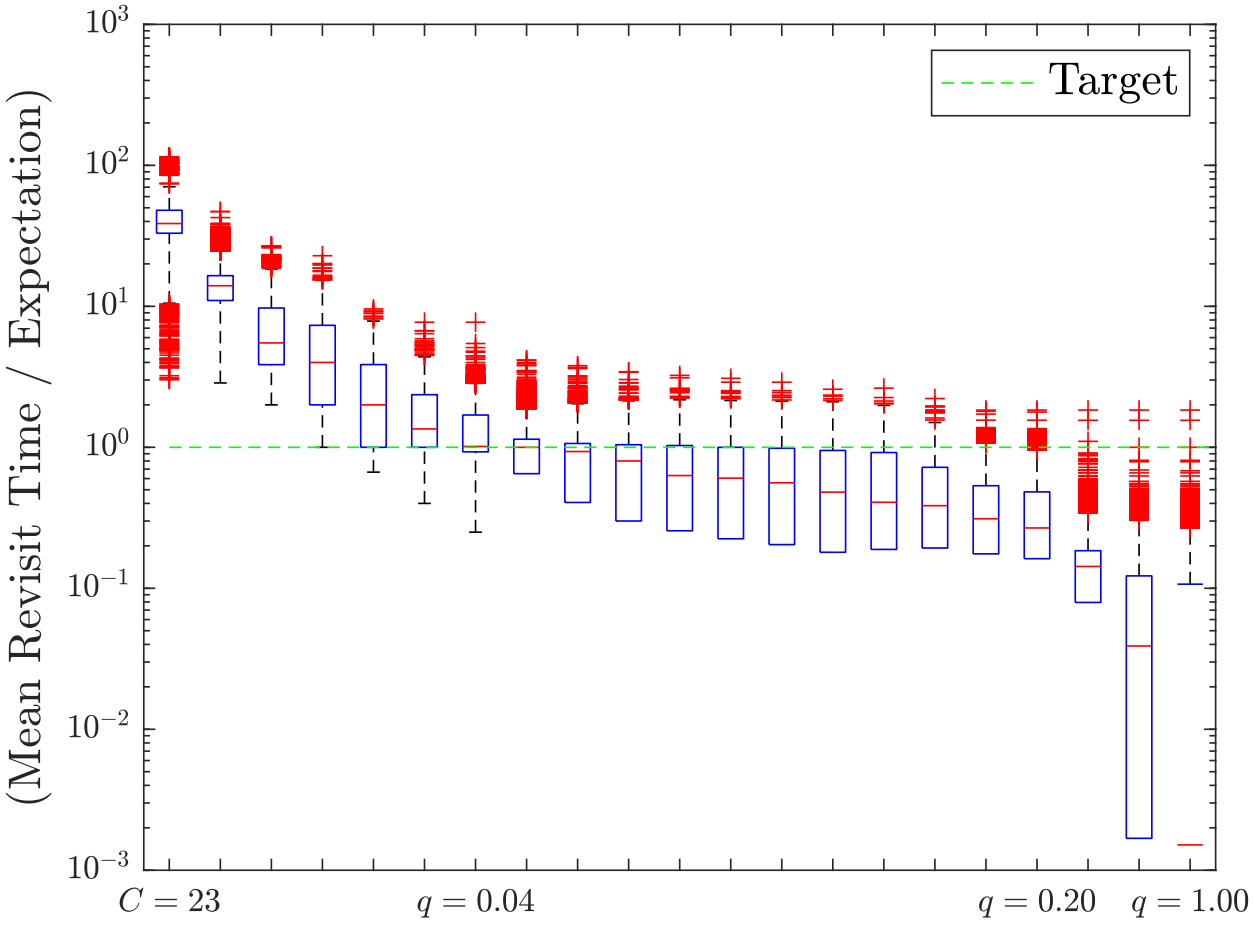

Figure 12. Collection of revisit times for each capacity test

The middle axis shows the same test results for the relative error of the GEO object, ψ_γ. The two subplots show similar results in that these two high interest objects reach relatively low error in this scheme by the end of the first day. Of course, the more sensing capacity, the quicker each object reaches a low error. In addition to this, in most tests both objects have negligible gains after the first day. The final subplot shows a heatmap of the results of all objects combined.

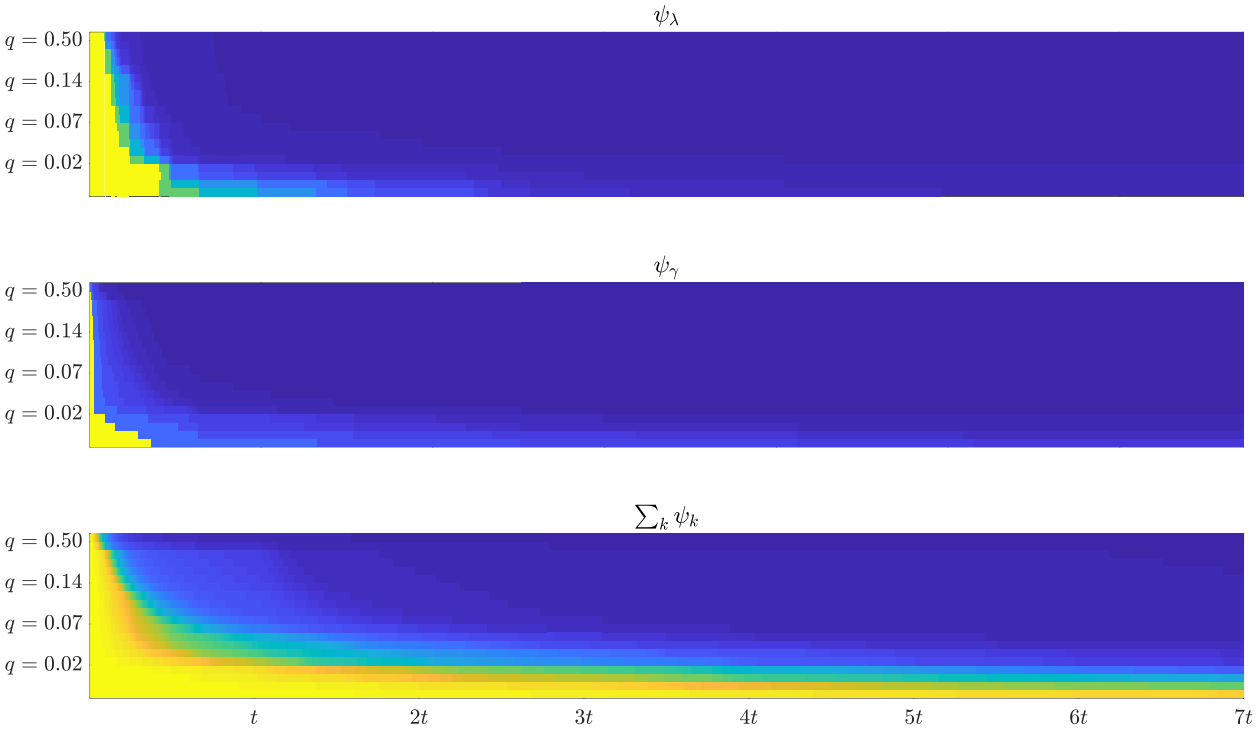

Figure 13. Total Positional Error over time heatmap

7 Results versus SP Tasker

The final step in this stage of testing is to provide evidence that the proposed tasking methodology, decision analysis, and overall collection scheme can compete in accuracy with SP Tasker. With all else considered, if accuracy between the two collection schemes are comparable then the other benefits of the SNARE architecture are added bonuses. Since SP Tasker is based off daily CTLs and not timesteps of less than a day, the GVF metrics will not be usable on SP Tasker unless they are bounded using heavy assumptions.

In this test, results will be compared to SP Tasker using data from Sensor Satisfaction (SenSat). SenSat is a data repository that contains information on sensor-satellite track requests and acquisition by the SSN. Formerly operated by the Joint Space Operations Center (JSpOC)–now the Combined Space Operations Center (CSpOC)–SenSat contains information from all the CTLs that SP Tasker produces daily such as tasking priority, the number of tracks and observations requested, as well as the actual count of tracks and observations returned by each sensor.

The exact same LVF function as the standalone tests will be used, as well as most of the testing conditions. In fact, the same visibility matrix, sensors, and RSO catalog are all considered. To eliminate as many biases as possible from the differences in the SenSat data, only the overlap between sensors and objects between the SNARE data and SP Tasker data are used, and the week simulated is the exact same week in time. The sensor overlaps still include opticals and ground-based radars, but each sensing component (if a site has multiple sensing components) are merged to a single consideration, so in this test $n = 15$. Evaluating which objects can be seen by which types of sensors, and calculating both regime and EDR group from the overlap of catalog objects between the SP Tasker data and the SNARE simulation, the observation expectation can then be formulated to real accuracy curves using SSNOII.

In this simulation, assume that the target accuracy for each object is the "optimal" point in the SSNOII curve (the "knee"), in which the number of additional tracks after the optimal quantity yields diminishing returns in accuracy. With this information, using general rules on the days per LUPI for each object (according to its EDR group), the appropriate count per day for each object is arrived to maintain catalog accuracy, $\hat{o}_k$. Because days per LUPI range between 6-28 for the RSOs, scaling tracks per LUPI down to tracks per day inevitably are not whole numbers. To support easier analysis of the results, each non-whole result is then rounded up. A distribution of the regimes and EDR groups are shown in Figure 14.

SSNOII curves are not available for every combination of inputs that might be used in the simulation, so some of the derived accuracy values are estimated using the curves that are available (this is discussed more thoroughly in Section 10). The $\hat{o}$ values then yield 1 for GEO objects, 5 or 8 for LEO objects, 2 for MEO objects, and 1 for HEO objects.

The final notable changes in this simulation against SP Tasker are the revisit and priority values. Since accuracy is the primary goal here, the revisit expectation is set to the reciprocal of the observation value and then rounded up to the nearest integer, $\hat{r}_k = \left\lceil \frac{1}{\hat{o}_k} \right\rceil$. Priority can be mapped to each object as well, using information from SenSat. Recall that SP Tasker bins the priority of each object into $p_k \in \{1, 2, 3, 4, 5\}$, and is dynamic with respect to the last time the object has been observed, among other criteria. Although it does not drastically affect the outcome, each priority

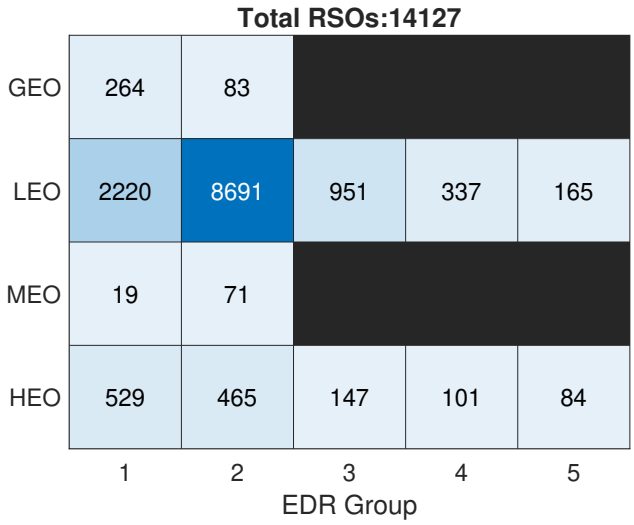

Figure 14. Distribution of RSOs per orbital regime and EDR group

value is computed using the statistical average over a chosen month of time for that object. In this case, each object then has a unique priority value.

Since this SNARE simulation is to be pitted against empirical results derived from SP Tasker, an actual measure of applied sensing capacity can be used to help determine properties of the system that were previously unknown. Not only can the SenSat data provide useful bounds on the available capacity between the 15 sensors, each particular sensor now has a realistic measure of how many objects it can track in a given day. Of course, these numbers are often affected by things like sensor down times and maintenance, and even weather in some cases so day-to-day there is some variability. Due to the potentially sensitive nature of these numbers, real capacity measures per sensor and the sensor network will not be explicit. However, percentages of the assumed network capacity will be explored, where a network capacity of "1" will refer to the historically determined sum of each sensor's individual daily track capacity.

7.1 Important Notes on SP Tasker versus SNARE

There are a couple key elements of the assumptions of this test environment that should not be ignored when reviewing the results, listed as follows:

- SP Tasker historically "over-tasks," which means both what SP Tasker is tasking to, and what SP Tasker receives as confirmed collects need to be considered separately.
- Though SP Tasker over-tasks on average, tracks accounted for in SenSat include serendipitous collects that cannot currently be simulated for the SNARE architecture.
- SNARE uses visibility assumptions based off what ATSAT can do, and SP Tasker uses a different program called Look Angle Module (LAMOD) to determine what it can see—only objects that could be collected in both SNARE and SP Tasker architecture are considered, though inevitably there will be variance on "possible collects" between the two paradigms

(ATSAT has been compared to LAMOD output to show the results are consistent in the aggregate but there may be isolated individual discrepancies).

- There is currently no way in the SNARE architecture to simulate for missed collections, in a way that is similar to SP Tasker over-tasking.

7.2 Results on the SNARE method

To remain consistent with the previous sets of tests, the normal metrics will be examined on the outcome of each test, varying with capacity. This may help illustrate differences in how the assumptions were formed between the standalone simulation and the SP Tasker-coupled simulation. Let $C_1 < C_2 < C_3 < C_\infty$. C_1 is the network track capacity just above the lower bounds of capacity assumed from SP Tasker during the simulation week. C_2 is just below the upper bounds of the assumed capacity of SP Tasker during the simulation week. C_3 is just outside the upper end of SP Tasker's empirical capacity of the week, but more closely reflects SP Tasker's capacity assumptions over a longer period of time. C_∞ represents the capacity if every single object that was visible at every timestep was collected, and is hence the bounds of what any decision process could achieve using the tools at hand. Results for C_3 are shown in Figure 15.

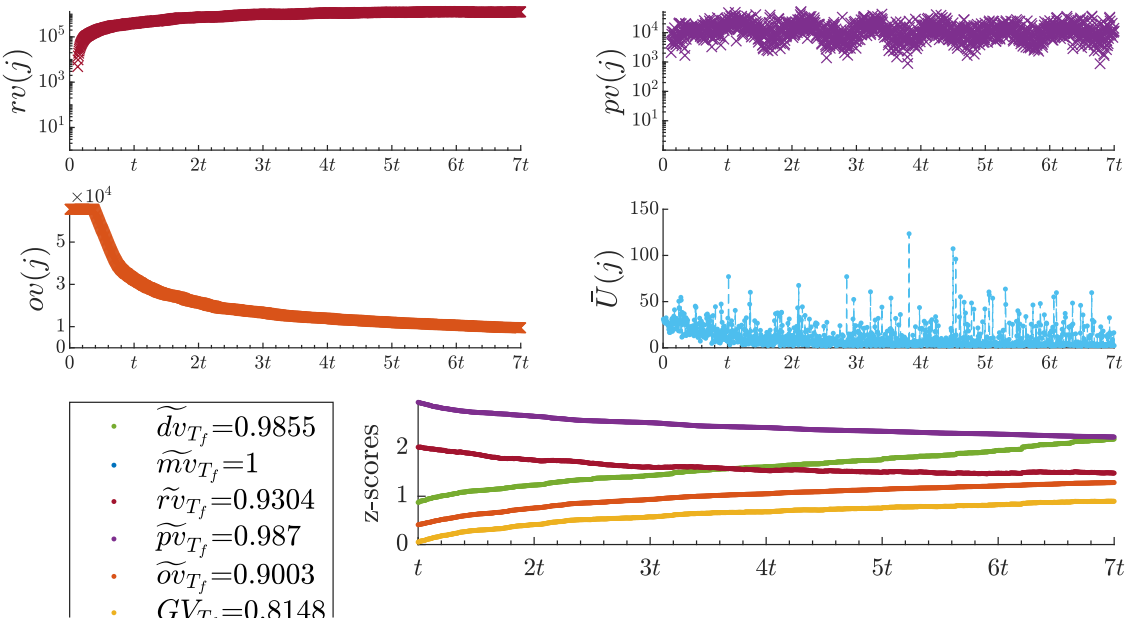

Figure 15. Metrics for C_3 over simulation time

In many ways the results of the SNARE logic under the assumptions formed for testing against SP Tasker mirror the results found under limited resource constraint in Section 6.2.2. The revisit violation count does continue to grow throughout the simulation but steadies off near the tail-end of the experiment (a longer simulation could be ran to show this more clearly, although that is outside the bounds of this study). Of course, in this scenario the revisit violations are heavily tied to the observation violations, which steadily decrease throughout the simulation and arrive close to a single value. Priority violations fluctuate throughout the simulation, maybe with some periodicity, but in general hover around a singular value. The average utility is unique from the other tests,

in that initially the utility of collect decreases near the floor, but then also has a high variance in the simulation, sometimes sporadically jumping up to a large number. This indicates that there are probably some RSOs that are sparsely seen in the visibility matrix and therefore largely increase their utility because they aren't seen per their expectations or are only seen by a particular sensor that is generally overloaded with high-value objects in its schedule.

Figure 16 demonstrates the test results of the SNARE logic using the modified assumptions.

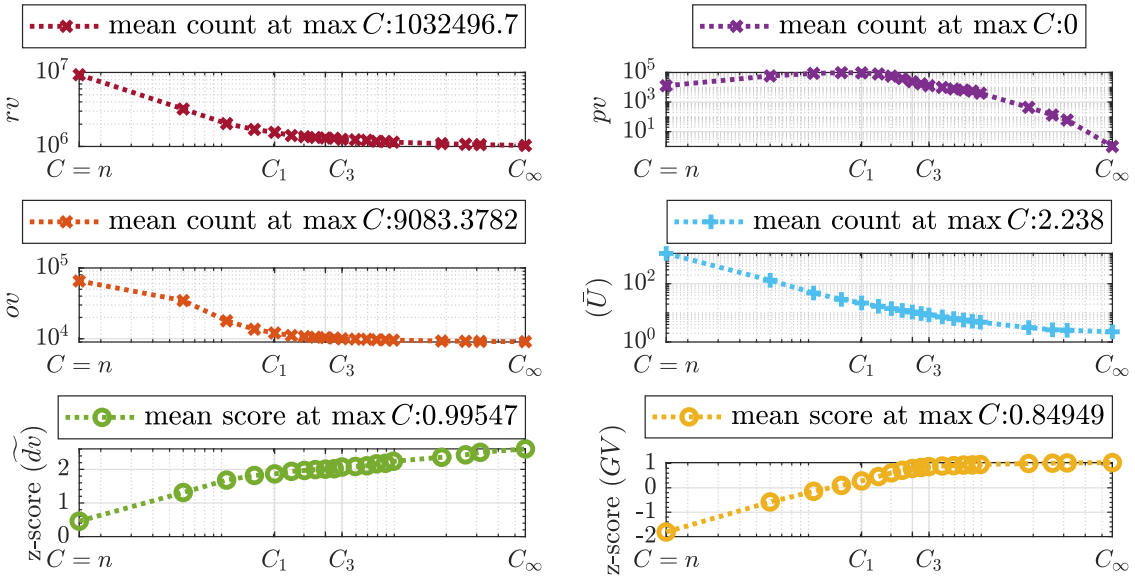

Figure 16. SNARE results under SP assumptions

The most interesting takeaway from the modified assumptions in attempt to match the SP Tasker environment, is the somewhat low score without any capacity restriction. A maximum global value score of only about 85% means that there is a disconnect somewhere between the set up of the test and the parameters of the objects. Of course, the priority violations do reach 0 with large enough capacity and the schedule diversity remains near perfect in an unlimited resource sense. This implies that the only two real factors contributing to the score at high capacity are the observation score and the revisit score (which is dependant on the observation expectation). Hence, even if everything that is visible (as generated by ATSAT) is captured, not everything can be seen often enough to satisfy the requirements.

7.3 Accuracy Results vs. SP Tasker

This section will discuss accuracy results against SP Tasker, looking at the differences between the two collection schemes regarding orbital regime/EDR groups as well as object sizes.

7.3.1 Regime and EDR Groups

A table illustrating key elements of the SNARE results in relation to the measured SP Tasker (received tracks) accuracy is in Table 2. A similar table illustrating key elements of the SNARE

results in relation to the accuracy of the expectation of SP Tasker is in Table 4 in Section 11. For both tables, the entries are unitless, because individually they are ratios of values in kilometers to another value in kilometers. In Table 2 (shown below), data from each column is the ratio of mean positional error to the corresponding mean error of received tracks, following direction from the daily CTL generated by SP Tasker for the given orbital regime and EDR group (assuming no loss). In other words, an entry in the table that has a value less than one implies that the accuracy measurement of that particular group of objects is more precise than the same group corresponding to SP Tasker's received accuracy, and a value greater than one implies the opposite. The first column labels which set of RSOs are being measured. The second column utilizes SenSat's data to calculate the number of expected tracks each object had over the set week of time, to which an accuracy measure can then be achieved over each group of objects. The third column is the "target" value that can be achieved at the knee of the curve in the SSNOII study. This value is what the CSpOC is tasking towards, i.e. it is the average desired accuracy of that particular set of objects. The fourth through seventh columns contain accuracy results from SNARE simulations, using the definitions from Section 7.2.

Table 2. Ratio of results versus SP Tasker received accuracy

Regime_Group	SP_accExp	knee	SNARE_C1	SNARE_C2	SNARE_C3	INF_acc
["GEO1"]	5.1568	0.2981	1.8271	1.6724	1.5019	0.7406
["GEO2"]	6.6260	0.4042	2.4381	2.3035	2.2938	0.9830
["LEO1"]	0.8098	0.2407	0.5387	0.3915	0.3555	0.1808
["LEO2"]	0.8387	0.1915	0.7427	0.6254	0.5911	0.4450
["LEO3"]	0.7551	0.1965	1.0163	0.8873	0.8370	0.6331
["LEO4"]	0.7571	0.1966	0.9315	0.7807	0.7210	0.4825
["LEO5"]	0.5609	0.2726	1.1736	0.9503	0.8600	0.5441
["MEO1"]	0.6101	0.6516	1.2007	1.1265	1.1255	1.0475
["MEO2"]	0.6861	0.7409	1.2702	1.1453	1.1414	0.8278
["HEO1"]	0.2520	0.7099	2.2862	1.6780	1.4020	0.5132
["HEO2"]	0.2902	0.7414	2.0495	1.6629	1.4096	0.4065
["HEO3"]	0.7878	0.9507	2.4658	2.2340	1.8151	0.3986
["HEO4"]	1.0568	1.5915	3.6248	2.7946	2.1775	0.5098
["HEO5"]	0.3874	1.6928	4.8394	3.8597	3.4247	0.8168
["ALL"]	0.8953	0.2716	0.9358	0.7715	0.7056	0.4331

Table 2 does show that accuracy always improves when network capacity increases, whether or not it meets the expectation set by the CTLs that SP Tasker produces. It would appear that objects in GEO are not over-tasked like some of the other objects. In fact, GEO objects seem to be under-tasked in relation to the knee accuracy shown in the third column, even though the collects on those GEO objects are significantly better than what is tasked. It is possible that this is an outcome of something like weather effects at a site level for those sensors that can see GEOs. Neither scheme meets the knee requirements, and in the SNARE construct it looks like it isn't

possible to collectively do so, even though at unreasonable capacities the accuracy that SP Tasker sets by collected tasks is achievable. The lower EDR group LEOs appear to be captured with better fidelity than SP Tasker, even at a low capacity assumption. Only EDR Group 1 has the potential to reach the knee accuracy for the LEOs using the visibility assumptions for this simulation. LEO EDR Groups 3 and 4 seem to reach SP Tasker's value with some of the higher capacity scheduling assumption, and SP Tasker expectations on LEO EDR Group 5 are not achieved by SNARE even though under infinite capacity assumptions that seems to be possible. Interestingly, SP Tasker expects high fidelity in each of these EDR groups for LEOs and does not receive that fidelity via confirmed tracks via its own direction.

The error expectation set by SP Tasker for MEO objects is lower than the knee in the curve, and although SP Tasker significantly over-tasks this group, SP Tasker's results are still far away from the knee value on average. SNARE also does not meet this criterion, and the average error from the desired amount is greater using SNARE logic, although using SNARE visibility logic assumptions it looks like it is not possible to meet the knee value across the catalog. HEO objects appear to be over-tasked even more so than with the MEO objects, and in all cases seem to be tasked to positional error well beneath what the tracks needed for knee accuracy would produce. Again, SP Tasker appears to have lower positional error than the SNARE logic on average. It would seem that

Table 3. Percent of objects per category that meet the knee standard

Regime_Group	SP_accExp	SP_accRec	SNARE_C1	SNARE_C2	SNARE_C3	INF_acc
["GEO1"]	7.95	82.95	81.82	83.33	83.33	86.36
["GEO2"]	7.23	87.95	81.93	84.34	84.34	86.75
["LEO1"]	16.35	8.24	5.27	37.66	54.68	90.05
["LEO2"]	8.55	5.60	2.57	16.49	27.68	63.31
["LEO3"]	27.44	7.68	0.53	5.68	10.73	37.33
["LEO4"]	19.29	10.09	0.30	3.56	8.31	42.43
["LEO5"]	46.67	23.64	1.82	6.67	14.55	58.18
["MEO1"]	73.68	5.26	21.05	21.05	21.05	26.32
["MEO2"]	42.25	14.08	9.86	15.49	15.49	64.79
["HEO1"]	95.27	68.24	72.78	80.15	82.80	89.60
["HEO2"]	95.27	68.17	75.05	80.86	84.30	92.69
["HEO3"]	93.20	74.83	80.27	82.31	83.67	96.60
["HEO4"]	94.06	87.13	83.17	88.12	89.11	97.03
["HEO5"]	95.24	92.86	72.62	77.38	77.38	92.86
["ALL"]	20.09	14.67	11.61	26.37	36.71	68.44

based off how SP Tasker is tasking, there are additional requirements (accuracy or other) making the tasks for HEOs higher than necessary and GEOs, for example, lower than needed. It could also be true that there is general a priori knowledge of the SSN that suggests that GEOs will be caught serendipitously and therefore don't need to be tasked as heavily, however this is only speculation.

It could also be the case that the accuracy measures are heavily skewed by a subset of objects in each group. The percentage of objects in each group that met or exceeded the knee requirements for accuracy are shown in Table 3.

Overall, there is at least one SNARE capacity assumption shown that exceeds the number of objects measured to meet the knee requirements in any given group, aside from those in GEO Group 2, LEO Group 4, and HEO Group 5. In the case of GEO Group 2 and even LEO Group 4 the SNARE construct does get fairly close, yet HEO Group 5 does get a considerable quantity of objects more than SNARE to the knee requirement (in this case achieving the limit of what the visibility assumptions allow SNARE to reach).

7.3.2 Object Sizes

Figure 17 demonstrates a log-log plot of the test results of the SNARE logic using the modified assumptions. The y-axis is the ratio between the SNARE logic collection scheme and the received accuracy by SP Tasker's collection scheme. Objects have been broken up into five separate bins using data from SenSat, and are plotted in orange with different black markers. The first bin, with objects under 5 centimeters in relative size has only 32 objects in it. The remaining bins, 5-10 centimeters, 10-50 centimeters, 50-100 centimeters, and 100^{+} centimeters have 3,106, 6,487, 1,418, and 3,033 objects in them respectively. The green line just represents the mark for each plot to beat to be more accurate than SP Tasker on average by the end of the simulation. Figure 23 is included in Section 11 and shows similar plots for the ratio of SNARE accuracy results versus SP Tasker's expectation, and the knee expectation.

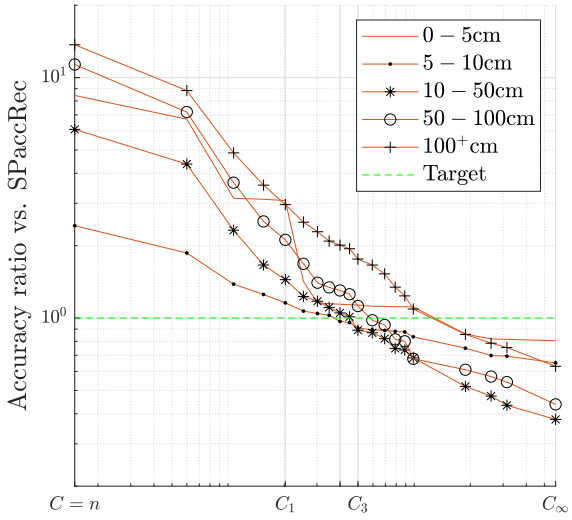

Figure 17. SNARE accuracy results for different object size bins

It is shown graphically that all accuracy results within the range of estimated SP Tasker capacity simulated by the SNARE construct are within 10 percent of SP Tasker's results. In particular, the SNARE construct might have a slight edge for catalog objects between 5-50 centimeters on average, whereas SP Tasker has a slight edge on objects smaller than that range, and objects 50-100 centimeters in size, as well as a slightly more sizeable accuracy gap on objects greater than a meter in size.

8 Conclusions

Several objectives have been defined within the context of SNARE performance and application. A methodology that maintains catalog accuracy levels to what they are today could be the minimal application of SNARE logic. In particular, SNARE logic provides an outline for a way to keep the catalog *at least* as accurate as what SP Tasker provides, but brings the pixelation of scheduling down from hours or days to perhaps mere minutes. The SNARE methodology is a construct that, in the moment, is primarily concerned with meeting revisit and observation expectations defined by the user, which outside of the war-fighting environment will provide catalog accuracy. When applied to the same catalog of objects inside a frame of war, there is the ability to detect and resolve changes within the governing system without destroying catalog accuracy in the process. The persistence of tracking objects in this paradigm leads to peace of mind given enough sensing resources, but also leads to SDA and increases true positional accuracy over time (which in turn could free up additional sensing resources).

8.1 Summary of Accomplishments

1. Claim–There exists a decision process that can be applied to accomplish the task of building a collection scheme usable to the SSN (LVF), which:
    - Can be pixelated to any valid time scale (primarily limited by sensor and communication timelines).
    - Is discretized using generalized qualities of sensors and RSOs through time.
    - Can be measured/monitored using well-defined metrics (GVF).
    - Can scale to any problem size.
    - Avoids tracking redundancies.
2. Claim–There exists a way to order value between sensor and RSO collections to provide direction for the decision process (utility function), which:
    - Accounts for unique object properties such as priority, revisit expectation, and observation expectation.
    - Accounts for physical properties between sensors and RSOs through visibility assumptions.
    - Rewards collection on RSOs that aren't being tracked frequently enough.
    - Rewards separated, and patterned collection on RSOs.
    - Considers interest level on objects without being greedy.
    - Excludes impossible collection pairings from consideration.
    - Assigns all valid collects some value, and is flexible enough to be extended to consider other values
3. Claim–There exists metrics that can monitor the goodness of a LVF to the application of building a collection scheme.

4. Claim–Through testing, it has been shown that the proposed decision process can generate a realistic collection scheme with minimal violations, utilizing tools that already exist.
    - The LVF was able to achieve schedule diversity, implying it is not a greedy process.
    - Observation and revisit expectations can be met, minimizing violations.
    - Accuracy can be measured using SSNOII.
    - The LVF methodology has been shown to work easily with serendipitous collects.
    - The LVF methodology has been shown to work easily with missed collections, through the employment of the bounty list.
    - The prioritization/selection methodology employed by the utility function is agnostic of sensor type.
    - RSO visibility/detectability can be determined using ATSAT.
    - The prioritization/selection methodology employed by the utility function is agnostic of RSO regime, EDR group, or size.
    - The collection scheme works well using TLEs.
    - Increased sensing capacity leads to a better collection, but there are better than linear returns on increasing capacity when under extreme resource load.
5. Claim–Through testing, it has been shown that SNARE might be able to maintain the catalog as well as or better than SP Tasker in general.
    - LEO objects in particular can be captured with greater fidelity, although GEO objects can be captured with similar fidelity, and a higher percentage of lower EDR group objects in HEO are able to mee the knee requirements.
    - Objects between 10 centimeters and 100 centimeters can be captured with slightly better accuracy, and all other objects can be captured at similar accuracy to SP Tasker.

At this stage in the design, SNARE has been able to provide valuable tokens of capability. The framework is already in place and utilizes tools that are trusted to provide a workable solution to the resource-loading problem of catalog maintenance and SDA. The logic can fit to any one application, or be tuned to behave differently for one reason or another.

8.2 Notes on Sensing Capacity

It is important to discuss the effects that sensing capacity has on the network. In 1994, the SSN was able to make 40,000 observations per day (tracking about 8000 catalog objects at the time), and that number has certainly grown since then[14]. However, in 1997 it was measured that the Ground-Based Electro-Optical Deep Space Surveillance (GEODSS) network of sensors, which is only part of the current SSN, was capable of generating up to 80,000 satellite observations per day by itself[15], and that was before the multiple sensor upgrades that the GEODSS network has received since 1997. And now, with a current catalog well into $20,000^{+}$ objects, the addition of a new space fence could shoot the catalog into seven digits. Regardless what the current SSN capacity is it should be realized that the need for maximum sensing capabilities is there. The

accuracy results acquired in Section 7.3 should clearly indicate that the desired accuracy of the catalog can be manipulated with different collection schemes, but also that any addition of sensing capacity can have a large effect on the network as a whole.

1. Effect of Capacity on Schedule Diversity

   It would appear that sufficient schedule diversity, regardless of object parameters, could be achieved at a collection rate of 4 percent of what is visible by each sensor, or about 125,000 tracks per day by the network. When SP Tasker was being prototyped, the target "success" rate of tracks distributed versus tracks received was 80 percent. This might indicate that the ability to have at least 150,000 tracks assigned for the given RSO catalog could be sufficient to see everything at least once per day.

2. Effect of Capacity on Schedule Revisit Time and TVT

   The revisit expectations set in the assumptions are deliberately strict. Because there isn't necessarily a current (unclassified) model of how revisit should be mapped to a given TLE, there are a lot of biases to the results simply due to construction. However, having a set time in between tracks on an object can significantly increase the positional accuracy achieved against the same number of tracks on an object that are spaced closely together and captured by the same radar. The quantification of these benefits is outside the scope of this paper, but it is at least obvious that can increase in capacity decreases TVT which supports the concept of persistence, which in turn supports the goals of the SNARE logic.

3. Effect of Capacity on Schedule Priority Violations

   Since priority violations are heavily tied to revisit violations, and represent a more abstract concept of schedule value, the most important thing to recover is that given enough sensor capacity, it has been shown that the priority violations go to zero. This satisfies one of the goals illustrated in Section 5.4, but in general gives reassurance that the SNARE construct does have some intuitive scalability. The point at which priority violations go away is probably the point at which the problem space no longer has a strict restriction on resources. The whole point of the Local Value function is to generally task objects that are higher priority (due to the slight correlation between observation and revisit expectations), but the power in the algorithm is that there are decisions made on a quick timescale that aren't exclusively biased towards object interest, and instead captures good network health.

4. Effect of Capacity on Positional Accuracy

   The positional accuracy of the catalog can be significantly increased with additional sensing resources. It might even be true that with the SNARE construct the potential gains are better than a linear increase in catalog accuracy with respect to gains in sensing capacity. The difference between the accuracy measurements recorded from the bounds of current capacity assumptions show that there is the potential to increase the number of objects meeting knee requirements to almost double the quantity that might be meeting the same requirement today. When it comes to objects like HEOs, that could mean that a few more optical or any type of deep space sensor can make a huge difference in catalog keeping using the SNARE algorithm.

8.3 Application to the SSN

In practice, the defined metrics can be used as tools to describe the environment and current state of the SSN. Average RSO utility can describe how resource constrained the system is, and particular utility metrics on any given object can be used to shed light on why the object gains value of collection. At any point in time, a quick count of objects not meeting accuracy requirements can be viewed by an orbital analyst. Or, if all objects are maintaining their accuracy requirements, it is reasonable to adjust the expectation values temporarily to increase catalog accuracy. Similarly, an idea of what happens to catalog accuracy can be derived from simulations where sensors go offline, lose capacity due to maintenance, or gain capacity due to sensor adjustment and tuning. The global score can be evaluated over time which can be correlated to both natural and political events which will lend itself to planning how to handle those events in the future.

The particularly helpful, and unique, nature of this applied methodology is its adaptability. Instead of trying to solve for an entire days' worth of collections at a single time, every timestep contains valuable information for how making the right decision for what needs to happen *next*. When unpredictable events occur, the SSN needs to be ready to change the plan instantaneously and waiting another day before sorting it out is rarely an option.

The simulation has handfuls of assumptions that affect the overall nature of what occurs from start to finish. However, it is very easy to change parameters and see how events unfold. The value function is easily tunable to achieve different results. To only increase catalog accuracy, f can be changed to reflect that desire, where f can be removed entirely or structured differently to favor the observation expectations. Similarly, in a war-fighting environment objects on the bounty list need to be more reactive to the value of collect, in which case α can be chosen to be arbitrarily large. The same officials that make decisions on which objects are more important than others can come to consensus in exactly how to use the LVF to achieve desired results of any shape.

A helpful product of the defined metrics could be some tool that gives insight to "problem" RSOs. They can be temporarily removed from consideration due to the amount of effort required to avoid violations, and those sensor resources can be allocated to objects that are deemed more important. In other words, if accuracy metrics on a particular object set needs to improve, there might be a way to optimally determine which less important objects can be dropped from the catalog.

8.4 Current and Future Work

Efforts are underway to fully design a test environment between SP Tasker and SNARE logic. The results in Section 7.3 are based off empirical data, and not data that changes in a real time environment. There is no way (in the current SNARE construct) to model missed tracks, true serendipitous collects, or any of visibility information other than a "yes" or "no" matrix generated by ATSAT using RCS and Signal to Noise Ratio (SNR) threshold values. However, the current effort to get SP Tasker and SNARE on the similar channels talking to Real Time Sensor Simulator (RTSS) will provide the ability to model events in real time, and simulate results that are more representative of real life.

Along with this, there is a desire to explore different ways to interface with each sensor in simulation. Because it is difficult to know how many objects a sensor can see in a given time step without

knowing which objects they are, some scheme is needed to figure out how many objects each sensor is tasked with to achieve results that are measurably better against SP Tasker. This could lead to an optimization method within that selection process that alters the LVF, but for initial testing the idea is to simply populate each sensor's task list until there is reason to believe it cannot support additional objects. This paradigm will never task redundant work unless there is no other work to do, and could potentially increase sensing efficiency which would lead to increased sensing capacity and therefore better results.

Once an interface is defined between each sensor, actual tasking lists can be dispersed to each sensor in the simulation, and then daily metrics would be run against SP Tasker. Knowing which objects are visible and at what time each simulated sensor decides to accomplish the tasks on their CTL, metrics in the GVF could be used to score SP Tasker. This will give us great insight on differences and similarities to both methods, and may help us further tune the collection method to get the best results possible.

9 Utility Function Analysis

Although there are many possible ways to devise a useful utility function within the construct of SNARE, the one that was chosen in the studies used in this paper satisfies all the criteria defined in section 5.2. Here, an analysis of some of the functional components of the utility function will be provided. The visibility component v will not be included in any analysis in this section (as the visibility component within this construct is a simple binary value). Recall the notional structure for the utility function (excluding the visibility component),

$$U = f(r;\hat{r}) * g(o;\hat{o}) * h(p)$$

where

$$f(r;\hat{r}) = 1 + \frac{r}{\hat{r}} * (\tanh(r - \hat{r}) + 1),$$
$$g(o;\hat{o}) = 1 + \frac{2 * \hat{o}}{2 * o + 1} * (\tanh(\hat{o} - o) + 1),$$
$$h(p) = \frac{p+1}{p}$$

The bounds for these functions were provided in 5.2 specifically for the parameters used throughout the simulations. General bounds where applicable will be examined in this section.

9.1 Analysis on f Component

An attempt at finding the analytical bounds of the f component will be provided here. Note that f is undefined when $\hat{r} = 0$, and $\lim_{\hat{r}\to 0^+} f = \infty$. For this reason it is necessary to constrain the revisit expectation of each object to be greater than 0. Hence, the functional evaluation on f will be conducted assuming that $\hat{r} \geq 1$ as in the problem definition. Similarly, the problem application constrains $r \geq 1$. Note that the function allows for $r \in [0, 1)$ but the following analysis would change.

The minimum of f occurs when $\hat{r}$ is sufficiently larger than r. This is generally true for any value of r but the minimum of f occurs at the minimum value of r. Fix $r = 1$. Let the minimum of f when $r = 1$ be denoted as $f_{\min}$. Then

$$f_{\min} = 1 + \frac{1}{\hat{r}} * (\tanh(1 - \hat{r}) + 1).$$

Recall that $\tanh x \in [-1, 1], \forall x \in \mathbb{R}$ and $\text{sign}(x) = \text{sign}(\tanh x)$. Hence, at maximum, when $\hat{r} = 1$, $f_{\min} = 2$. Otherwise, when $\hat{r} > 1$, $1 < f_{\min} < 1 + \frac{1}{\hat{r}}$. Therefore

$$f_{\min} \in (1, 2].$$

Furthermore, if the problem definition constrains the revisit expectation to never exceed the number of timesteps in a given day (i.e. every object must be seen at least once per day), then a definitive bound can be determined

$$f_{\min} \in \left[\frac{1 + t + \tanh(1 - t)}{t}, 2\right],$$

for $\hat{r} \in [1, t]$.

Because the tanh function is bounded, the dominating value of f is determined by r. In other words, this function will begin to linearize as r becomes sufficiently larger than $\hat{r}$, and f will grow to infinity–there is no $f_{\max}$. However, if it is assumed that every object *is* seen once a day, then $\hat{r} \in [1, t]$. Then $f_{\max}$ is greatest when the revisit expectation is the strictest. Fix $\hat{r} = 1$.

$$f_{\max} = 1 + r * (\tanh(r - 1) + 1)$$

and when $r \in [1, t]$,

$$f_{\max} = 1 + t * (\tanh(t - 1) + 1).$$

And, because $\sup(f_{\max}) = 1 + 2t$,

$$f \in [f_{\min}, 1 + 2t).$$

Now, let $r \geq 1 \in \mathbb{R}$ and $u_f = f(r)$ be a continuous function in the real domain, with parameter $\hat{r} \geq 1 \in \mathbb{R}$. Consider the derivative of function u_f with respect to input revisit interval r using the chain rule,

$$u'_f = \frac{r * \sec^2(r - \hat{r}) + \tanh(r - \hat{r}) + 1}{\hat{r}}$$

Note that $\sec^2 x > 0, \forall x$ and $\inf(\tanh x) = -1$, hence $u'_f > 0, \forall r, \hat{r}$. Therefore, f is increasing monotonically (*strictly*) on its domain when $r \geq 1, \hat{r} \geq 1$. In other words, as the input r increases for a chosen $\hat{r}$, the newly obtained value of f is always greater than all values prior.

9.2 Analysis on g Component

An attempt at finding the analytical bounds of the g component will be provided here. Note that g is undefined when $o = -\frac{1}{2}$. However, in the context of the problem, negative observations are not viable and hence is automatically constrained by the problem such that $o \geq 0$ and the functional evaluation on g will be conducted under that assumption. As in section 9.1, it will be assumed that at least one track is desired per day, so let $\hat{o} \geq 1$ (technically a similar analysis as is provided would work for any $\hat{o} > 0$ without loss of generality). Additionally, within the constraints of the problem definition it is reasonable to bound o to be as large as there are timesteps in a day, since more tracks than timesteps implies multiview violations.

The minimum of g occurs when o is sufficiently larger than $\hat{o}$. This is generally true for any value of o but the minimum of g occurs at the maximum value of o. Fix $o = t$. Let the minimum of g when $r = t$ be denoted as $g_{\min}$. Then

$$g_{\min} = 1 + \frac{2\hat{o}}{2t+1} * (\tanh(\hat{o} - t) + 1)$$

Recall that $\tanh x \in [-1, 1], \forall x \in \mathbb{R}$ and $\text{sign}(x) = \text{sign}(\tanh x)$. Hence, at maximum, when $\hat{o} = t$, $g_{\min} = 1 + \frac{2t}{2t+1}$. Otherwise when $\hat{o} = 1, 1 + \frac{2}{2t+1} * (\tanh(1 - t) + 1)$. Assuming $t > 1$,

$$g_{\min} \in \left[1 + \frac{2}{2t+1} * (\tanh(1 - t) + 1), 1 + \frac{2t}{2t+1}\right]$$

and for sufficiently large t,

$$g_{\min} \in (1, 2)$$

Because g does not grow with increasing o, as long as $o > 0$, g is bounded and there is a $g_{\max}$ and it occurs when there are no tracks within the last day. However, to avoid contradicting assumptions in 9.1, fix $o = 1$ (i.e. a single track is assumed to have been gathered within the last t timesteps).

$$g_{\max} = 1 + \frac{2\hat{o}}{3} * (\tanh(\hat{o} - 1) + 1)$$

and when $\hat{o} \in [1, t]$ (such that the track expectation on o is reasonably bounded to avoid multiview violations),

$$g_{\max} = 1 + \frac{2t}{3} * (\tanh(t - 1) + 1).$$

And, because $\sup(g_{\max}) = 1 + \frac{4t}{3}$,

$$g \in [g_{\min}, 1 + \frac{4t}{3}).$$

Now, let $o \geq 1 \in \mathbb{R}$ and $u_g = g(o)$ be a continuous function in the real domain, with parameter $\hat{o} \geq 1 \in \mathbb{R}$. Consider the derivative of function u_g with respect to input track count o using the chain rule and quotient rule,

$$u'_g = -\frac{2\hat{o}(2(\tanh(o - \hat{o}) + 1) + (2o + 1)\text{sech}^2(\hat{o} - o))}{(2o + 1)^2}$$

Note that $\text{sech}^2 x > 0, \forall x$ and $\inf(\tanh x) = -1$, hence $u'_g < 0, \forall o, \hat{o}$. Therefore, g is decreasing monotonically (*strictly*) on its domain when $o \geq 1, \hat{o} \geq 1$. In other words, as the input o increases for a chosen $\hat{o}$, the newly obtained value of g is always less than all values prior.

9.3 Analysis on h Component

An attempt at finding the analytical bounds of the h component will be provided here. Without the need for any particular assumptions, it can be seen that with any RSO catalog of size m, (regardless of whether there are m distinct priority levels), the bounds of this component lie within (1,2]. In fact, $\forall p \in \mathbb{Z}^+, h \in (1, 2]$. More explicitly,

$$h \in \left[\frac{p_{max} + 1}{p_{max}}, \frac{p_{min} + 1}{p_{min}}\right]$$

and in context to the problem definition, where $p = [1, \ldots, m]$,

$$h \in \left[\frac{m+1}{m}, 2\right].$$

Note that $h > 1$ if $m > 0$.

Let $p \geq 1 \in \mathbb{R}$ and $u_h = h(p)$ be a continuous function in the real domain. Consider the derivative of function u_h with respect to input priority level p using the quotient rule,

$$u'_h = \frac{p - (p+1)}{p^2} = -\frac{1}{p^2}.$$

Note that u'_h is negative $\forall p$ and hence is monotonically decreasing (*strictly*) on the domain of h. In other words, as the input p increases, the newly obtained value of h is always less than all values prior.

9.4 Summary of Analysis and Implication

Given the problem definition the following constraints have been placed on the system:

- It is assumed that every object is seen at least once per day.
- It is assumed that every object is seen at most once per timestep, and hence seen at most t times in a day.
- It is assumed that every object in a catalog of length m has a distinct, ordinal, priority level.

Which leads to the following variable/parameter bounds:

- $r \in [1, t]$
- $\hat{r} \in [1, t]$
- $o \in [1, t]$
- $\hat{o} \in [1, t]$
- $p \in [1, m]$

And the functional components of U have the following properties under the domain of each bounded variable/parameter:

- $f \in [f_{\min}, 1 + 2t]$, where $f_{\min} \in (1, 2]$ and f is monotonically increasing
- $g \in [g_{\min}, 1 + \frac{4t}{3}]$, where $g_{\min} \in (1, 2)$ and g is monotonically decreasing
- $h \in [\frac{m+1}{m}, 2]$ and h is monotonically decreasing

A figure of the relative range of contributions from f, g to U is shown in Figure 18 and a 3-D version, 25 is included in Section 11. It then follows that the sensitivity of each functional component and its corresponding input variable relies heavily upon the parameter selection associated with each input variable. In other words, it could be the case that variations within r for a selection of $\hat{r}$ could cause f to have more effect in U than g for variations within o for a selection of $\hat{o}$ (or vice versa). It is unlikely that h would have more of an effect on f, g on U since the range of h falls within the general range of the minimums of f, g.

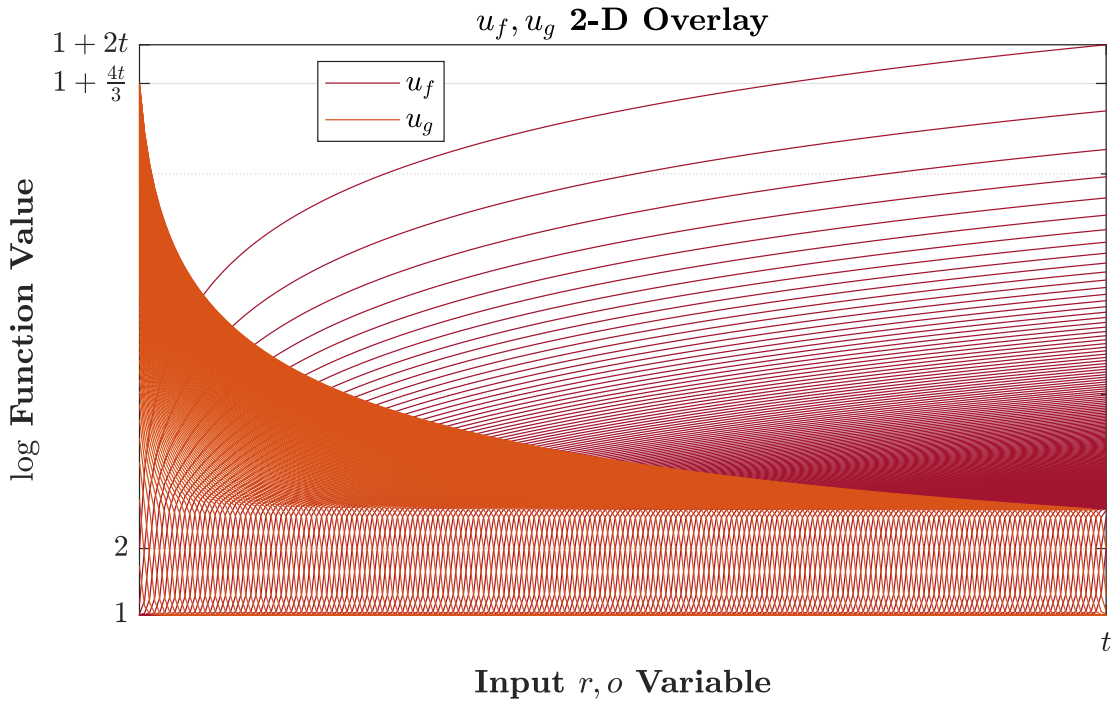

Figure 18. Possible contributions to U from f,g over domain $[1, t]$

It isn't general practice to perform sensitivity analysis on closed-form functions of simple complexity (especially if any of the inputs are unbounded), but now that the inputs have been bounded with practical assumptions and there is uncertainty surrounding the prominence of the range of its components, a basic analysis on the sensitivity of the inputs to the chosen utility function can be done by treating the utility function as a "black box."

9.5 Sensitivity Analysis on Utility Function

Let $X = \{r, \hat{r}, o, \hat{o}, p\}$ with bounds as determined in section 9.4. The sensitivity analysis on U will be performed using a Maximum Difference Method by varying input X to the function B, where $U = B(X)$.

In this method, sensitivity analysis will be performed by evaluating the amount the bounds on U change when the full range of any input parameter component of X is varied. The basic methodology is as follows:

- A single element of X is chosen as the "test subject."
- All other elements of X are chosen within their appropriate range at random.
- The test subject is varied between all integer values in $[\min X, \max X]$ (unless the test subject is p in which case it is varied linearly between $[1, 2]$) and all other randomly chosen elements of X are held constant.
- The range of U is measured along the domain of X.

This is repeated equally among all elements of X, with 10000 trials each. As an additional constraint, since values of r, o have an implication of each other, it is assumed that $o + r \leq t + 1$. Essentially, it doesn't make sense to say that $r = 192$ if $o = 192$ following the assertions in Section 9.4 (if $o = 192$, then $r \leq 1 \rightarrow r = 1$).

The result of this test is a distribution of $U_e = B(X)$ curves, with $e \in X$ being the active (varied) element of input parameter X. Let $R_e = \max U_e - \min U_e$. Then R_e describes the distribution of total variation of U_e by varying input parameter X with active element e. Let $S_e = \sum R_e$ over every test run. S_e is then essentially the area of the distribution of the range of U_e with input X and active variable e. This is the quantity over each element of X that is shown in Figure 19.

It is shown that the r counter and $\hat{r}$ parameter (and hence u_f) has more contribution to the overall value of U than the o counter and $\hat{o}$ parameter. However, it is also shown that under these constraints that the u_g component still has a fair weight on the value of U, and even on the same order of magnitude as u_f. As intuition would suggest, the u_h component has the least weight on the overall value. It is likely that the selection of the ranges of values for each variable has a large effect on the sensitivity of the system as a whole, but this set of ranges is fairly reasonable and at least gives us an idea about what to expect over the span of a normal day. Recall that almost regardless of parameter selection that $u_g >> u_f$ when r is low and o is low and that $u_f >> ug$ when r is large (over a large enough time frame this is true regardless of the other variables/parameters).

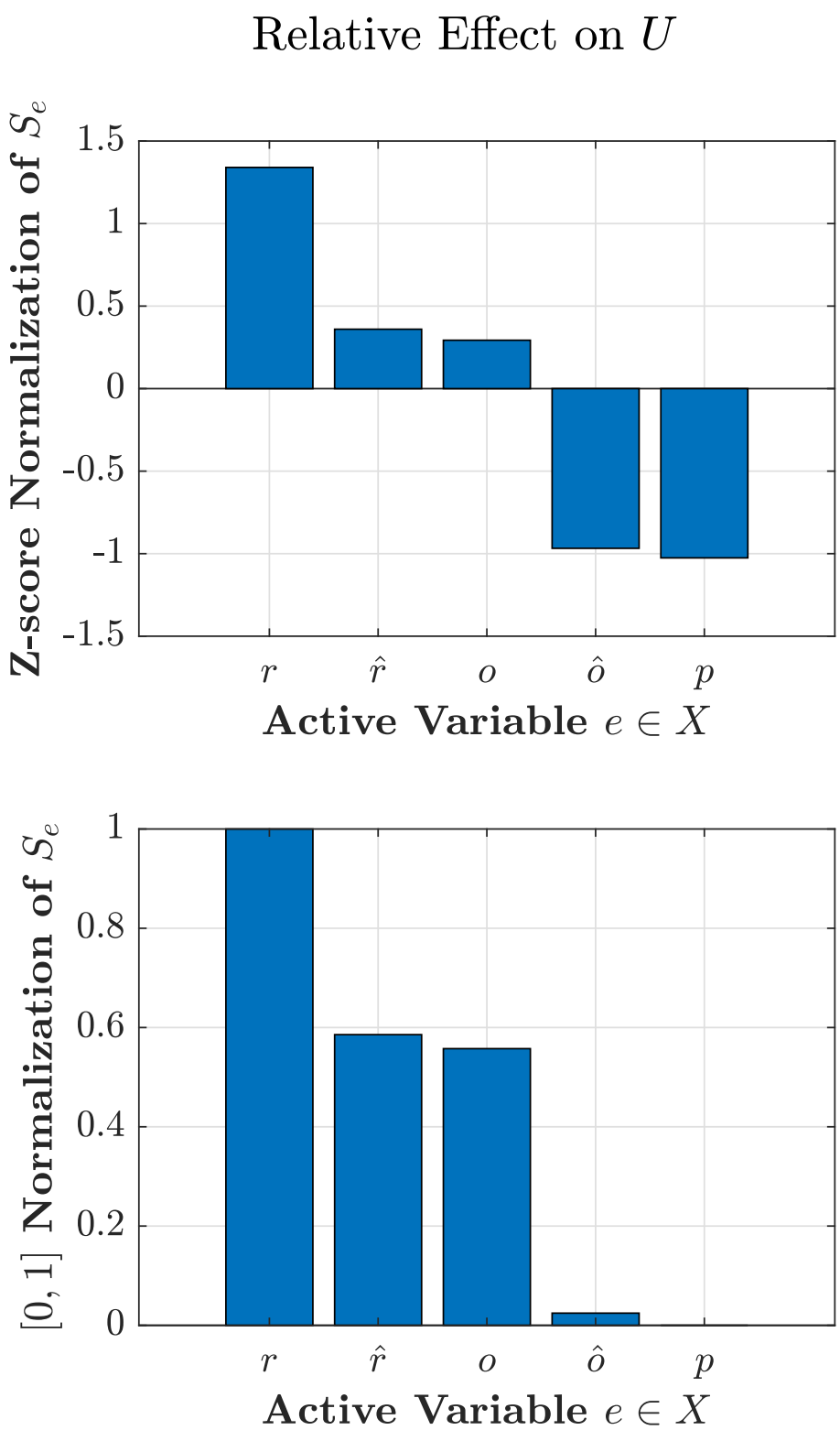

Figure 19. Sensitivity Analysis results

10 Notes

This section is meant to be a resource to provide expansion for certain technical concepts in the paper.

10.1 Artificial Intelligence

The value functions were initially designed with artificial intelligence in mind. The use of a hyperbolic tangent function is commonly used in machine learning methods, where tanh is used as an "activator" or "activation function." This is an appealing concept in many senses, because the system would be highly reactive to important events, but otherwise could make logical linear decisions. It is actually likely, given enough test data, that this problem could be solved using a machine learning algorithm. Every object in space that is not under the control of a human being should have a predictable, calculable path. Given geography of the sensors in a network, and the geometries of paths that each RSO takes, it is not unreasonable to have a self-sustaining network of sensors that "know" which objects they can generally see. In fact, patterns may even emerge out of serendipitous collections that can help improve sensor models or understanding of orbital properties of objects in space using machine learning. The future possibilities on this are endless, but for the time being without having any proof that sensors can be effectively tasked on events happening in the near future on a basic algorithmic design, it is something that is a little out of reach.

Initially, An agent-based Particle Swarm Optimizer (PSO) method was considered, but could not objectively be proven that it was better than the LVF decision logic.In fact in many cases, the objective method used by SNARE was much quicker to achieve similar results to what the PSO offered and is deterministic opposed to stochastic (this fits better into future iterations of the construct). However, that is how the design of the metrics were formed. The idea of needing a set of scores to determine actions provided the necessary metrics and utilities that the SNARE method currently operates under (both LVF and GVF). It is not unreasonable to go back to this method (or similar methods) in the future, but having a deterministic outcome for now is the best option.

10.2 SSNOII

SSNOII tables have been very important in the development of testing between SNARE and SP Tasker. It is an easy tool to use that helps give perspective on tangible catalog accuracy between two completely different collection schemes. However, there is some incompleteness in the SSNOII tables. Given the distribution of RSOs in Figure 14 and the variability of track mixes in the SNARE versus SP Tasker experiment, there are actually multiple SSNOII input entries that do not yield numerical data. For example, only EDR Group 1 is available for MEOs and GEOs. The only reasonable workaround for the MEOs and GEOs that are in the catalog as EDR Group 2 objects, are to approximate the Group 2 objects using Group 1 curves. In addition, LEO objects of any group do not have curves for using only ground optical sensors, and only LEO Groups 1-2 have available curves for a joint track mix of radar and ground opticals. Hence, when applicable, the SSNOII curves that were referenced were approximated. If a LEO object was only seen using a ground optical, and was a Group 1 or 2, the mixed sensor curves were used, and otherwise the

appropriate group under the assumption of being seen only by radar were used. Interestingly, the HEO curves are complete for all groups and viable track types, so the positional accuracy for HEO objects was always appropriately calculated.

11 Additional Figures and Tables

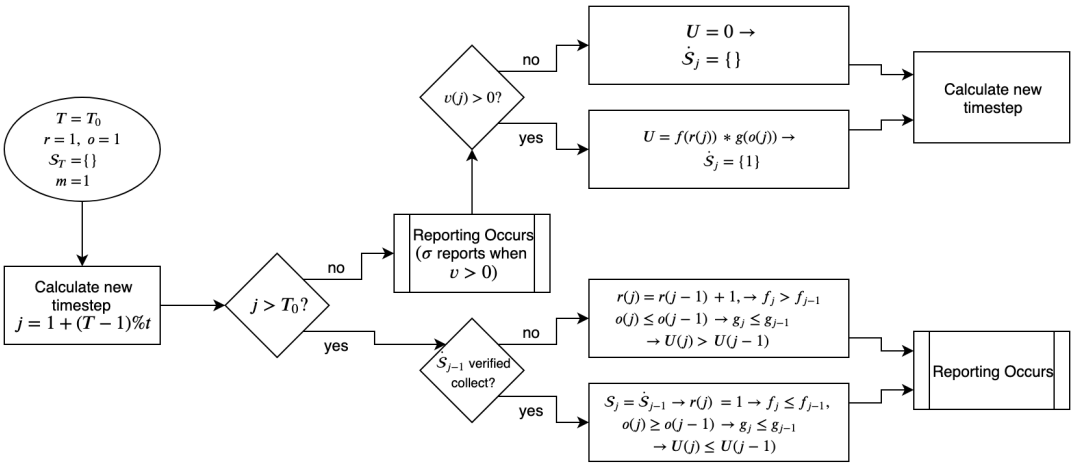

Figure 20. Pseudo code base case diagram

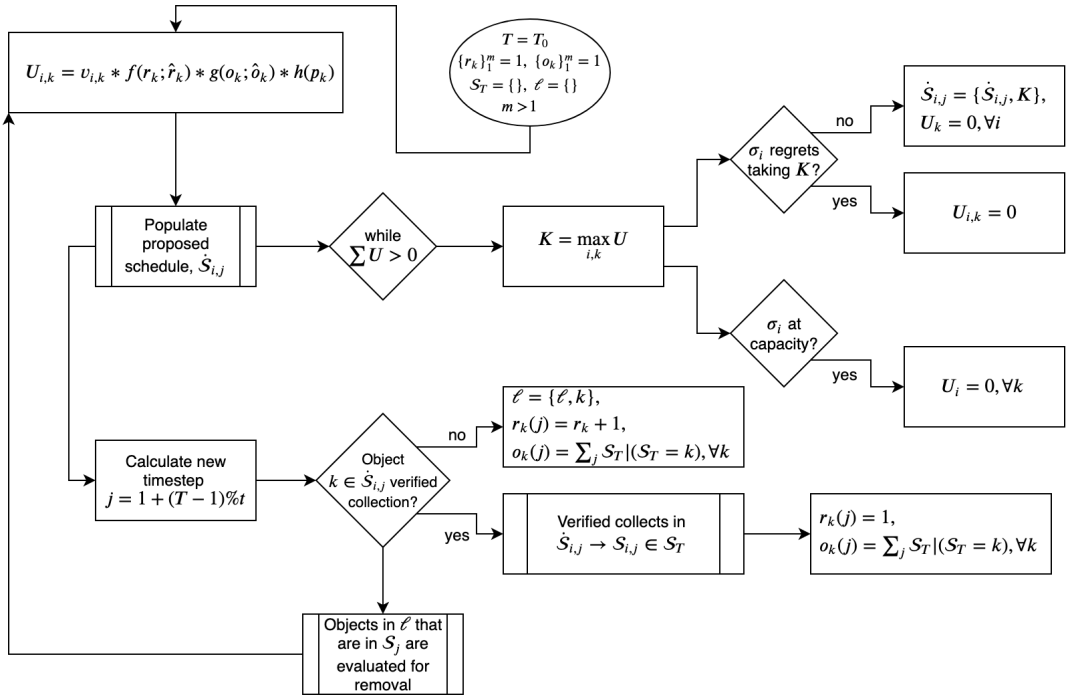

Figure 21. Pseudo code base case diagram

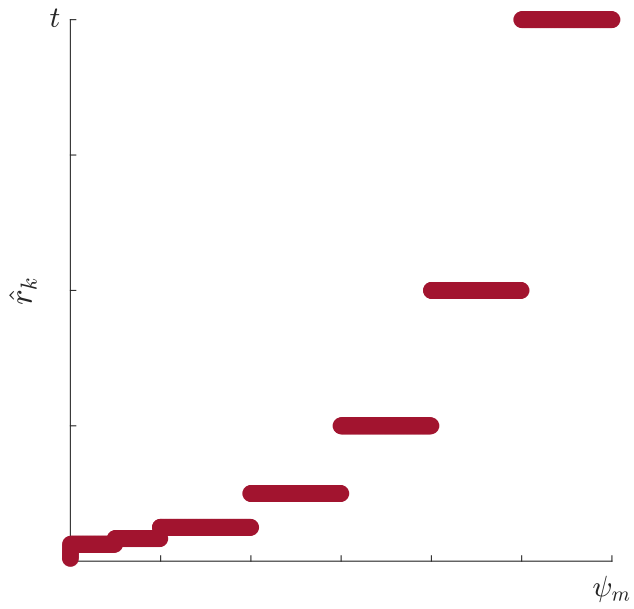

Figure 22. $\hat{r}$ expectations for all objects in the simulation

Table 4. Ratio of results versus SP Tasker expected accuracy

Regime_Group	SP_accRec	knee	SNARE_C1	SNARE_C2	SNARE_C3	INF_acc
["GEO1"]	0.1939	0.0578	0.3543	0.3243	0.2912	0.1436
["GEO2"]	0.1509	0.0610	0.3680	0.3476	0.3462	0.1484
["LEO1"]	1.2349	0.2972	0.6652	0.4835	0.4390	0.2233
["LEO2"]	1.1923	0.2284	0.8855	0.7457	0.7048	0.5306
["LEO3"]	1.3243	0.2602	1.3459	1.1751	1.1085	0.8384
["LEO4"]	1.3209	0.2597	1.2304	1.0312	0.9523	0.6373
["LEO5"]	1.7829	0.4860	2.0923	1.6942	1.5333	0.9702
["MEO1"]	1.6391	1.0680	1.9681	1.8465	1.8449	1.7170
["MEO2"]	1.4575	1.0798	1.8514	1.6692	1.6636	1.2066
["HEO1"]	3.9679	2.8169	9.0713	6.6580	5.5629	2.0363
["HEO2"]	3.4461	2.5551	7.0628	5.7307	4.8578	1.4009
["HEO3"]	1.2694	1.2067	3.1300	2.8357	2.3041	0.5060
["HEO4"]	0.9462	1.5059	3.4299	2.6444	2.0605	0.4824
["HEO5"]	2.5816	4.3702	12.4935	9.9643	8.8413	2.1088
["ALL"]	1.3804	0.4638	1.5180	1.2228	1.0956	0.6000

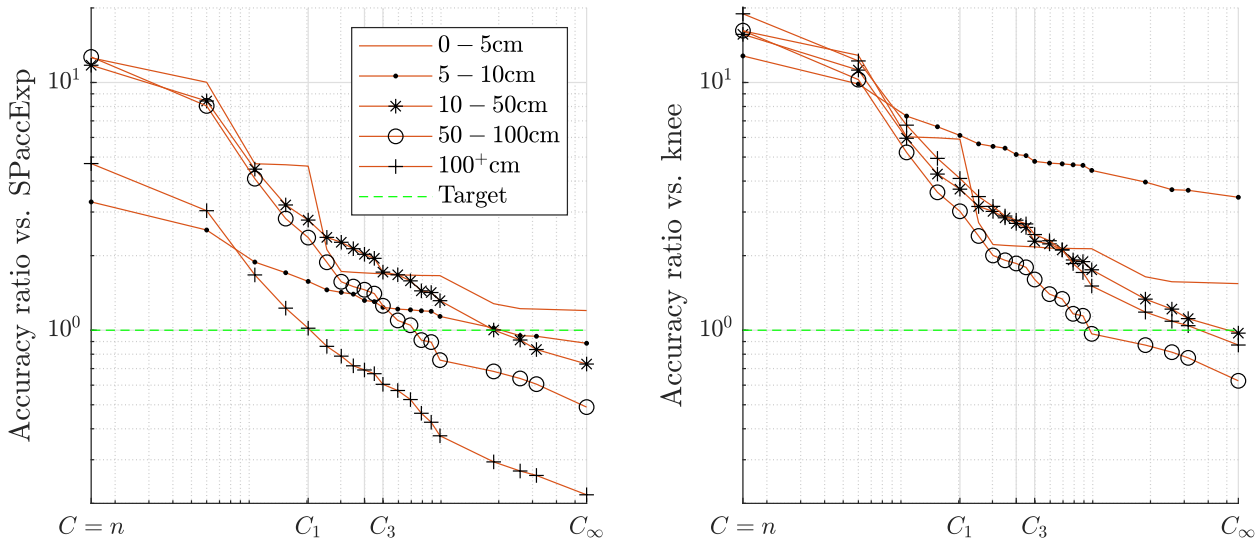

Figure 23. SNARE accuracy results for different object size bins

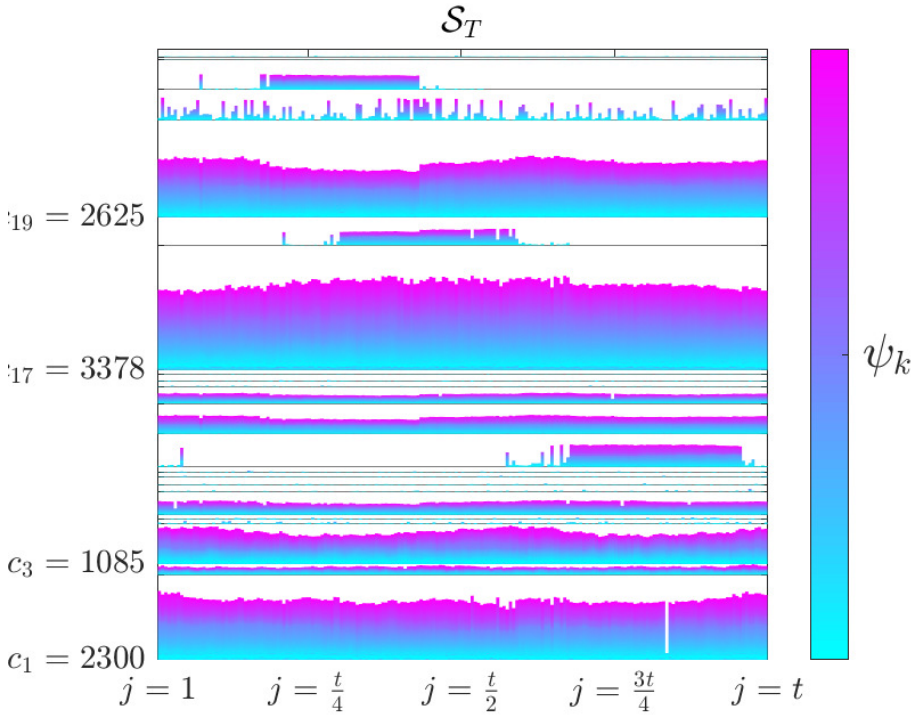

Figure 24. Example Visual Representation of $\mathcal{S}_T$

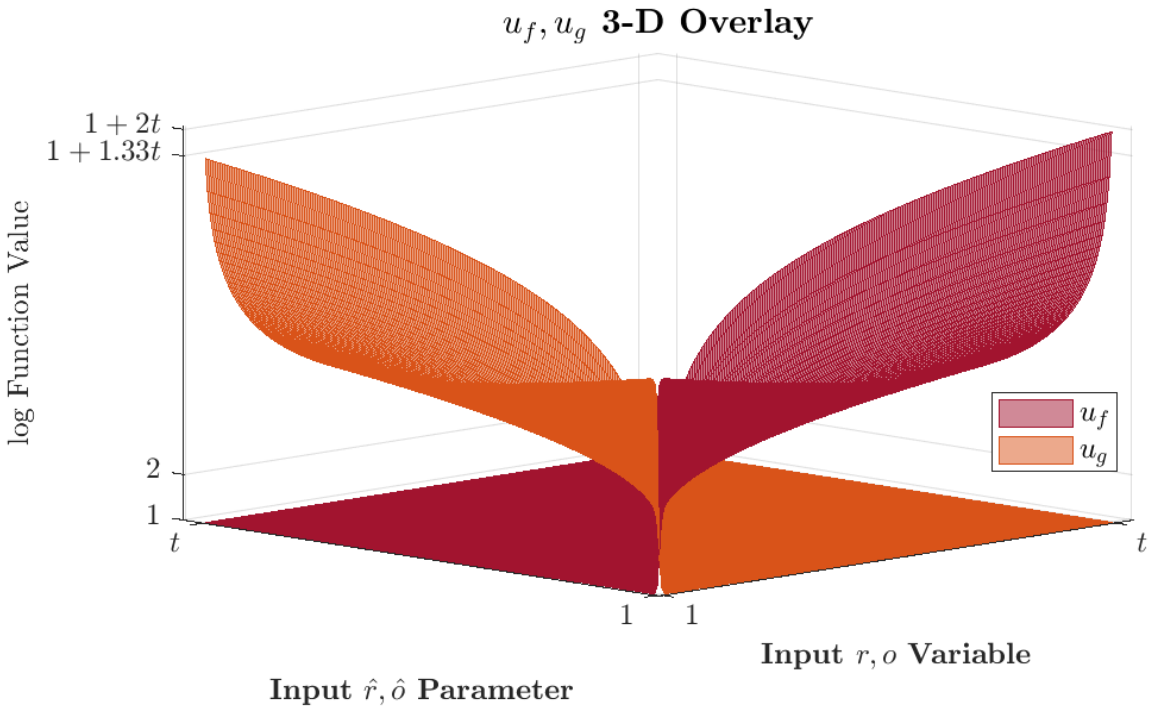

Figure 25. Possible contributions to U from f,g over domain $[1, t]$

12 References

[1] T. Berg. (2018, December) Irvine students launch second satellite in a month. [Online]. Available: https://www.nasa.gov/mission_pages/station/news/orbital_debris.html

[2] J. Green. (2019, April) Satellite built by virginia tech undergraduates ready for launch into space. [Online]. Available: https://vtnews.vt.edu/articles/2019/04/news-aoe-cubesat.html

[3] E. Mahoney. (2017, August) First cubesat built by an elementary school deployed into space. [Online]. Available: https://www.nasa.gov/feature/first-cubesat-built-by-an-elementary-school-deployed-into-space

[4] L. Wood. (2018, June) Global space industry market and technology forecast to 2026: A $558 billion opportunity, driven by demand for nano-satellites and re-usable launch vehicle systems. [Online]. Available: https://www.businesswire.com/news/home/20180628005882/en/

[5] E. W. Tate Ryan-Mosley and K. Kakaes. (2019, June) The number of satellites orbiting earth could quintuple in the next decade. [Online]. Available: https://www.technologyreview.com/s/613746/satellite-constellations-orbiting-earth-quintuple/

[6] Y. Takeuchi, "Space traffic management as a guiding principle of the international regime of sustainable space activities," *Journal of East Asia and International Law*, vol. 4, no. 2, November 2011.

[7] U. COPOUS, "Long-term sustainability of outer space activities," *Report of the scientific and Technical Committee on its fifty fourth session*, vol. A/AC.105/2017/, no. CRP.26, June 17.

[8] DONALD J. TRUMP, "Space Policy Directive-3, National Space Traffic Management Policy," https://www.whitehouse.gov/presidential-actions/space-policy-directive-3-national-space-traffic-management-policy/, 2018, online; Issued on: June 18, 2018.

[9] ——, "Space Policy Directive-4," https://media.defense.gov/2019/Mar/01/2002095015/-1/-1/1/SPACE-POLICY-DIRECTIVE-4-FINAL.PDF, 2019, online; Issued on: February 19, 2019.

[10] J. G. Miller, "A new sensor allocation algorithm for the space surveillance network," *74th MORS Symposium*, vol. 5, August 2006.

[11] M. Hejduk and R. Frigm, "Collision avoidance," NASA/CNES, Short Course 1, September 2015.

[12] *North American Aerospace Defense Command Cheyenne Mountain Complex Integrated Tactical Warning and Attack Assessment Program Technical Report-Study Services Special Perturbations Tasker Programmer Documentation for the Integrated Space Command and Control (ISC2)*, Lockheed Martin Information Systems and Global Solutions, 9970 Federal Drive Colorado Springs, CO 80921-3616, July 2014.

[13] L. Newman, "Space Surveillance Network Optimization II (SSNO II)," Tech. Rep.

[14] B. L. Petrick, "Weighting scheme for the space surveillance network automated tasker," Master's thesis, Air Force Institute of Technology Air University, January 1994.

[15] D. T. Kelso, "Space surveillance," *Satellite Times*, vol. 4, no. 1, September 1997.

[16] M. Garcia. (2015, April) Space debris and human spacecraft. [Online]. Available: https://www.nasa.gov/mission_pages/station/news/orbital_debris.html

A List of Variables

Topic	Variable	Brief Description
SSN notation	n	number of sensors
	σ_i	i^{th} sensor in the SSN
	c	theoretical timestep track capacity of a sensor
	C	theoretical timestep track capacity of the SSN
	D	theoretical daily track capacity of the SSN
RSO notation	m	number of RSOs
	ψ_k	k^{th} RSO in the catalog
	p	priority level of an RSO
	r	revisit count of an RSO
	$\hat{r}$	revisit expectation of an RSO
	o	observation count of an RSO
	$\hat{o}$	observation expectation of an RSO
Collection notation	$\dot{\mathcal{S}}$	proposed collection scheme
	$\mathcal{S}$	verified collection scheme
	$\mathcal{H}$	serendipitous collect list
	$v_{i,k}$	visibility of object k by sensor i
LVF notation	$U_{i,k}$	the utility of collect of object k by sensor i
	ℓ	the bounty list
	α	describes how quickly U_k changes if $k \in \ell$
GVF notation	GV	global value score
	dv	diversity metric
	mv	redundancy metric
	rv	revisit metric
	pv	priority metric
	ov	observation metric
	$\mathcal{D}$	set of RSOs in a collection (elements have multiplicity of 1)
	$\mathcal{P}_k$	set of RSOs in a collection with lower priority than object k
	$\widetilde{mv}$	the multiview score
	$\widetilde{rv}$	the revisit score
	$\widetilde{pv}$	the priority score
	$\widetilde{ov}$	the observation score

This page intentionally left blank.

B Acronyms

ATSAT Architecture Trades and Sensor Assessment Tool.

CRD Capstone Requirements Document.

CSpOC Combined Space Operations Center.

CTL Composite Tasking List.

DoD Department of Defense.

EDR Energy Dissipation Rate.

ELSET Element Set.

GEO Geo-stationary Earth Orbit.

GEODSS Ground-Based Electro-Optical Deep Space Surveillance.

GVF Global Value Function.

HEO Highly Elliptical Orbit.

JSpOC Joint Space Operations Center.

LAMOD Look Angle Module.

LEO Low Earth Orbit.

LTS Long-term Sustainability.

LUPI Length of Update Interval.

LVF Local Value Function.

MEO Medium Earth Orbit.

PNT Positioning, Navigation, and Timing.

PSO Particle Swarm Optimizer.

RSO Resident Space Object.

RTSS Real Time Sensor Simulator.

SDA Space Domain Awareness.

SenSat Sensor Satisfaction.

SID Satellite Identification.

SNARE Sensor Network, Autonomous, Resilient, Extensible.

SNR Signal to Noise Ratio.

SP Special Perturbation.

SPADOC Space Defense Operations Center.

SPD Space Policy Directive.

SSA Situational Space Awareness.

SSN Space Sensor Network.

SSNOII Space Surveillance Network Optimization #2.

STM Space Traffic Management.

TLE Two-Line Element Set.

TVT Total Violation Time.

UN COPUOS The United Nations Committee on Peaceful Uses of Outer Space.

VT Violation Time.