Performance Analysis of Algorithms for Coordination of Earth Observation by CubeSat Constellations

Performance Analysis of Algorithms for Coordination of Earth Observation by CubeSat Constellation

Andrew K. Kennedy* and Kerri L. Cahoy†
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

DOI: 10.2514/1.I010426

Two algorithms that coordinate activities across a resource-constrained, Earth-observing CubeSat constellation are presented. The Resource-Aware SmallSat Planner algorithm performs online planning of activities for a satellite while keeping the satellite's resources within constraints. The Limited Communication Constellation Coordinator algorithm performs coordination of observations across the constellation to reduce average revisit times for a set of targets. The algorithms are simulated for 24 h with an 18 satellite LEO constellation. Three orbital geometries are examined, with different configurations of intersatellite and satellite-to-ground communications links to share planning information. Results indicate that coordination through Resource-Aware SmallSat Planner/Limited Communication Constellation Coordinator and a background communications constellation improves observation performance, with sensor-averaged revisit times of 197, 203, and 225 min (over three orbital geometries) versus 204, 211, and 240 min for a baseline random sensor selection method. The results also reveal that the constellations studied perform poorly at sharing planning information via only downlinks and crosslinks and point to the need for a method of calculating the information sharing utility of such links in a coordinated constellation. These findings are relevant for Federated Satellite Systems because they provide guidance on next steps toward integrating large networks of heterogeneous small satellites into large-scale, coordinated, Earth-observing systems.

I. Introduction

IN THIS paper, we assess the feasibility of automated onboard coordination of Earth observations across a constellation of resource-constrained CubeSats. We analyze the utility of coordinated observations while varying both the constellation's orbital geometry and the use of communications links for sharing planning information. This work is relevant for the Federated Satellite Systems community in that it takes a step toward the integrated planning of Earth remote sensing and communications link usage, two key attributes of future Earth-orbiting satellite systems. We assess the results from two newly developed planning algorithms and describe important items of future work for a full-scale planning system.

Traditional Earth-observing space missions use a single, monolithic spacecraft to take measurements and communicate with ground stations. This architecture limits geospatial and temporal spacing of an instrument's measurements because it provides only a single space-time location for them to be taken. Large-scale, Earth-orbiting constellations offer great benefits due to the additional spatial and temporal diversity they provide. Such large constellations are rapidly emerging as a real possibility with the advance of small satellite and CubeSat technology. A CubeSat is a class of small satellite built of multiple $10 \times 10 \times 10$ cm units each with a mass around 1.3 kg and designed to a standardized launch vehicle interface [1]. The CubeSat form factor imposes significant limitations on operations; they are generally very resource- and communications-constrained, with tens of watt-hour batteries [2,3], tens of watts of solar-cell production [4], and low-gain antennas with low data rates [5]. This is in contrast with larger satellites, which typically use hundreds to thousands of watts of power and have masses of hundreds to thousands of kilograms. The constraints imposed by the CubeSat platform suggest there are significant benefits to increased coordination and careful planning.

To increase the effectiveness of a CubeSat constellation for Earth observation, we incorporate coordination in the scheduling of individual satellites' onboard activities. In this work, we consider a constellation of CubeSats with tightly coupled intersatellite planning, in which each satellite performs online (i.e., real-time), onboard planning of its own activities, while maintaining onboard energy and data resource constraints within bounds. The satellites use planning information obtained via communications links with other satellites and ground stations to inform their own choices about observation timing. In this work, the goal of observation planning is to reduce the revisit (interobservation) times achieved across the constellation, for a predefined set of regions on the Earth's surface. Other metrics may be used to judge performance, but this revisit time metric is useful for initial assessment. This tightly coupled operations planning scheme offers several potential benefits, including: 1) the ability to closely align measurements from multiple satellites in space and time, 2) faster response to spontaneous observation opportunities, 3) more effective routing of data to the ground to improve data latency, and 4) robustness to faults on individual spacecraft.

The problem of single-satellite Earth observation and communication planning has been extensively investigated in the literature, often with a focus on deciding which observation tasks to execute and how to best meet the myriad timing and priority constraints between observations [6–8]. But these algorithms are not necessarily directly applicable to the CubeSat domain because of the need to manage limited onboard resources. Other work has addressed this [illegible]ct by also considering resource utilization in the planning process. For example, the Automated Scheduling and Planning Environment (ASPE[illegible]cheduling system generates initial schedules and then uses a technique called iterative repair to modify the plans such that they meet desired constraints, including those on resources [12]. The system, including its onboard replanning c[illegible]erpart Continuous Activity Scheduling Planning Execution and Replanning ([illegible]PER), has been deployed on the Intelligent Payload E[illegible]ment CubeSat [13]. Spangelo and Cutler use a linear programming formulation to schedule downlinks while satisfying resource constraints [14], and Monmousseau details both a simulated annealing and mixed-integer linear programming (MILP) approach for an augmented problem that addresses observation and communications scheduling in a constellation of CubeSats [15]. In our work, we use a similar MILP-based formulation to perform onboard planning but restrict the planning to a single satellite.

Received 30 September 2015; revision received 7 July 2016; accepted for publication 1 August 2016; published online XX epubMonth XXXX.

*Graduate Student, Department of Aeronautics and Astronautics, Building 37-354, 77 Massachusetts Avenue.
†Assistant Professor, Department of Aeronautics and Astronautics, Building 37-354, 77 Massachusetts Avenue. Member AIAA.

Various architectures have been investigated in the literature for multisatellite cooperation. Several researchers proposed architectures that require little ground involvement in the planning process. Surka et al. [16] and Schetter et al. [17] discuss a multisatellite cooperative architecture based on "expert agents" running onboard. The system was planned to be deployed on the TechSat-21 mission, augmented by a set of resource-aware automated planning algorithms designed by Chien et al. [18,19]. The overall system provides significant operational independence but requires the satellites be in a highly connected network. Das et al. [20] as well as Van der Horst [21] and Van der Horst and Noble [22] investigate task allocation methods that use crosslinks between the satellites and require less network connectivity. These approaches help reduce the need for frequent communications with the ground but sacrifice schedule quality across a widely distributed constellation because they do not directly attempt to optimize plans for the whole constellation. The issue of schedule quality is better handled by planning systems with a centralized, usually ground-based component that can consider the whole constellation in the planning process. Monmousseau handles this through a MILP algorithm running on the ground, with plans strategically uplinked to satellites [15]. Damiani et al. discuss a system with satellite onboard activity planning and a centralized ground station system that distributes observation requests to satellites [23]. Yet these architectures do not consider the utility of integrating intersatellite crosslinks into the distribution of planning information across the constellation.

There has also been a great deal of work on multi-agent cooperation outside of space applications. The "tBurton" factored planner extends the classical planning approach to planning for interactions between multiple agents by reasoning about the temporal interdependencies of their tasks, and exploiting hierarchy to reduce the planning complexity [24,25]. Decentralized partially observable Markov decision processes can be used to find an optimal, decentralized plan across multiple agents by explicitly reasoning about uncertainty in the agents' performance of activities [26,27]. Gombolay, Wilcox, and Shah implemented a set of algorithms using a MILP formulation to quickly allocate and schedule sets of tasks to multiple robots on a factory floor. Market-based task allocation algorithms use interagent bidding on tasks based on the agents' local valuation of the tasks [28]. These approaches tend to break down in the constellation application, however, because of their need for relatively frequent communications between all agents.

It is clear from the literature that several algorithms are available for the scheduling of cooperative observations by an Earth-observing constellation. Yet none of these approaches fully addresses the problem of how best to augment centralized planning with intersatellite crosslinks given the inherently limited communications and resource availability of a CubeSat constellation. In this paper, we take a first step toward such a solution by developing a system that enables the incorporation of information sharing communications links in the planning process. The work makes three contributions: 1) the development and application of two algorithms, Resource-Aware SmallSat Planner (RASP) and Limited Communication Constellation Coordinator (LCCC), for distributed, online prioritization of observations across a constellation of resource-limited CubeSats using crosslinks for information sharing; 2) an assessment of these algorithms' coordinated observation performance using representative communications and planning strategies; and 3) an assessment of the effectiveness of sharing planning information across the constellation using representative communications strategies. These contributions serve as a base reference for future work toward a layered planning system incorporating both ground-based and onboard planning that fully quantifies and optimizes the usage of information sharing links.

The rest of this paper is organized as follows. Section II provides context for the algorithms by introducing a concept of operations for the coordinated constellation and giving a high-level introduction to the algorithms' role. It also covers some of the assumptions of the developed simulation and the metrics used for performance assessment. Section III describes both the Resource-Aware SmallSat Planner (RASP) algorithm, which is used for activity planning on a single satellite, and the Limited Communication Constellation Coordinator (LCCC) algorithm, which handles planning information sharing across the constellation. Section IV provides details on the parameters for the specific simulation cases run and the satellites' resource and activity constraints. Section V discusses the results obtained from the simulation cases. Section VI concludes with a recapitulation of the contributions of this work, a discussion of the limitations of the algorithms, and items of interest for future work. Note that this work synthesizes and expands on findings in previous work [29–31] and presents a full, coherent application of the RASP and LCCC algorithms to a coordinated CubeSat constellation.

II. Constellation Simulation Context

The following subsections introduce the models used for the constellation and its satellites and provide context for the algorithms detailed in Sec. III. The first subsection introduces the operational model for a single satellite. The second and third discuss the architecture of the algorithms and the communications links used for coordination across the constellation. The fourth illuminates the important assumptions of the models, and the fifth introduces the performance metrics used for this work.

A. Small Satellite Operational Model

A simple but practical operational model is used for the CubeSats in the constellation. This model is represented as a state-transition diagram for planning purposes, as shown in Fig. 1. This diagram shows all the possible transitions between activities or operational modes onboard the satellite. The spacecraft can only perform one activity at a time.

There are three main types of activity that each satellite individually schedules in its onboard planning process. Coordination activities are affected by other satellites in the constellation. These include observations, where the satellite observes a desired region of the Earth's surface. The beginning and end of an observation are determined by when the subsatellite point crosses into or out of a specified latitude/longitude area for a region (no field-of-view considerations are added for the instrument in this version of RASP and LCCC). Observations are planned by each satellite while also considering the utility of observations already planned by other satellites. Each observation uses one of a set of three sensors, A, B, and C, that represent three different onboard scientific instruments. Crosslinks and commlinks are another coordination activity, during which two or more satellites communicate with each other via intersatellite radio link. During a crosslink, two satellites in the constellation communicate. During a commlink, a satellite in the constellation talks to an external satellite (such as in the Iridium constellation). These link types are explained in more detail in Sec. II.C.

Resource management activities are used to manage two types of onboard resources, energy storage (ES) and data storage (DS). ES refers to the energy stored in onboard batteries, which is depleted during all activities (except recharge) through the use of onboard electronics and hardware. In this model, ES is recovered during the recharge activity, a dedicated mode during which the satellite orients itself so as to maximize input power. We assume a dedicated recharge mode because of the power-intensive observation mode; adaptation to a more general model is an item for future work. DS refers to the data stored in onboard persistent memory, which is collected as engineering telemetry during all activities, and additional science data during the observation. Downlink is used to reduce DS by sending data to a ground station via a radio link. Limits are enforced on ES and DS, as detailed in Sec. IV.C and Table 3. The transition activities are used to move between the other modes. Slew occurs when the spacecraft uses its attitude control subsystem actuator suite to change its attitude. Idle is a general, low-resource usage state entered when no other activities are being performed. We assume every activity produces or consumes each resource (ES and DS) at a constant rate, as detailed in Sec. IV.D. The availability and timing of windows for observation, downlink, crosslink, and recharge activities is obtained from a simulation of the satellite's orbit, as detailed in Sec. III.A.1.

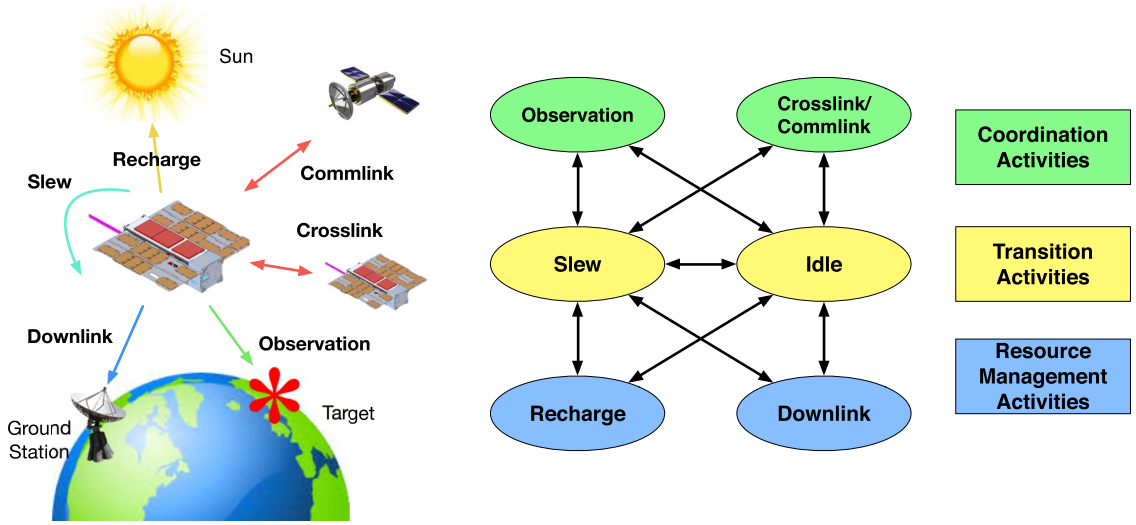

Fig. 1 State-transition diagram representation of the operational activities for a single CubeSat.

B. Constellation Autonomy Architecture

A constellation of satellites is constructed, all operating as dictated by the operational model in Fig. 1. Two algorithms, the Resource-Aware SmallSat Planner (RASP) and the Limited Communication Constellation Coordinator (LCCC) are run onboard the satellites. The algorithms' goal is to minimize average revisit times for the regions observed by all the satellites in the constellation. The average revisit time metric is defined in Sec. II.E. Other metrics exist for judging observation performance, such as percent coverage of the Earth's surface or degree of geometric overlap in measurements. Average revisit time serves as a good first assessment of the algorithms' performance. To achieve good revisit time performance, the satellites share information between each other about their planned observation timings and use this information in turn to plan their own sequences of activities. A satellite's shared planning information simply constitutes a list of future observation activities to be performed by the satellite, including start and end times, the region to be observed, and the sensor type to be used.

The RASP algorithm performs low-level planning for each satellite, choosing the activities to perform at each instant in time to both maximize observation performance and maintain ES and DS within desired limits. RASP uses the shared planning information to weight the importance of any given observation and then chooses which observations to perform and which sensors to use for them. The LCCC algorithm is run in a distributed fashion across all satellites in the constellation; in essence, it is a protocol for updating shared planning information. When satellites communicate with each other, they gain access to new planning information and update their own database of information about all satellites' planned observations. This database is then used by RASP in the low-level planning process. The relationship between RASP and LCCC is depicted in Fig. 2 [32]. RASP is run at a regular interval onboard each satellite, and LCCC governs the sharing of information between the satellites.

C. Communications Links

There are three types of communications link that each satellite uses: downlink, crosslink, and commlink.

Downlinks are used to both downlink stored data to the ground and share observation planning information with the ground (using simultaneous uplinks). The satellites downlink through a widely distributed network of nine ground stations, as shown in Fig. 3 (with details in the Appendix).

Crosslinks are used by the satellites to share planning information directly between each other. The crosslinks can be performed using a simple, low-data-rate, "isotropic" radio link, readily achievable with commercial off-the-shelf hardware for CubeSats [5]. Link budget calculations with representative hardware indicate that a 9600 bps crosslink can be achieved at intersatellite distances up to 2400 km. We assumed a transmitter output power of 4 W, 0 dB gain for both the transmitting and receiving antenna (isotropic assumption), a frequency of 450 MHz, polarization loss of 4 dB, and a system noise temperature of 126.8 K. Based on our calculations of the size of stored planning information in bytes, the crosslink must last about 6 s to share planning information for an 18-satellite constellation.

Commlinks give the satellites access to a commercial low Earth orbit (LEO) communications constellation, such as the Iridium or Globalstar constellations [33,34]. This is helpful for simulating a situation where the satellites have regular access to a backbone communications network. The onboard commlink radio is assumed to also be omnidirectional and have the same parameters as the one used for crosslink, except that access to the backbone constellation is available from any point in LEO.

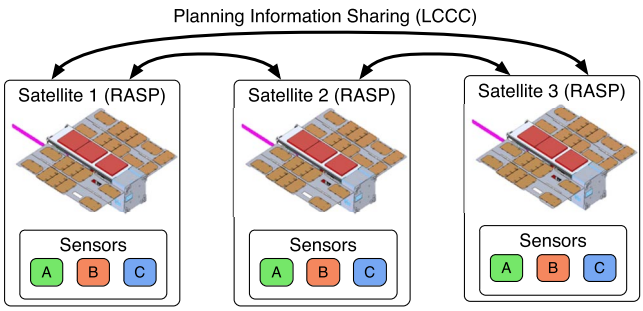

Fig. 2 Framework used for automated coordination across the constellation. RASP handles activity planning onboard each satellite; LCCC coordinates information sharing between them (image modified from Kennedy and Cahoy [32]).

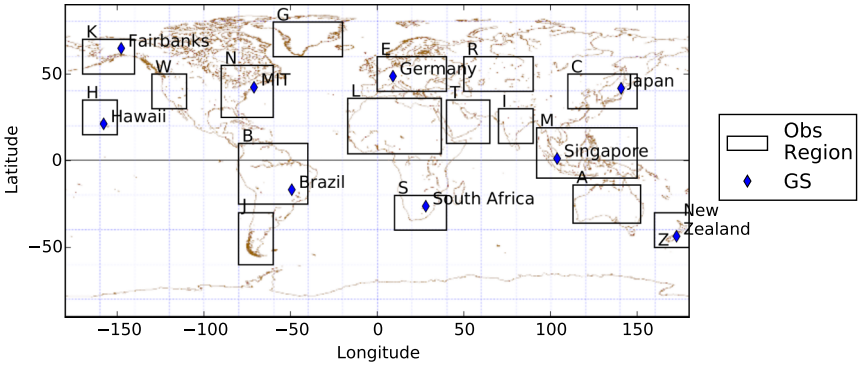

Fig. 3 Regions (17 total) and ground stations (nine total) used in constellation simulations.

Note that, for a real mission implementation, it would be necessary to perform a cost-availability analysis for both crosslinks and commlinks. Both links would require some cost for implementation and radio frequency licensing. In this analysis, we only consider the performance of the links with regard to planning information.

D. Model and Algorithm Assumptions

There are several important assumptions made in this work, which we now identify specifically to aid in understanding the relevance and limitations of the algorithms presented.

In our current model, we assume that a satellite is only capable of performing a single activity at a time (e.g., it cannot both observe and crosslink at the same time). This assumption was made primarily to reduce planning complexity. As a result, the algorithms cannot schedule overlapping observations. Also, observation events are modeled in a simple way. Possible observations are identified by when a satellite passes over a rectangular latitude–longitude region on the Earth's surface. We chose to use regions of interest on the surface instead of field of view as this parameter is often application-specific and may depend on orbit geometry and attitude.

We assume crosslinks can only occur when satellites approach within 2400 km of each other. Actual planning information exchange can only happen when both satellites simultaneously perform the crosslink. We do allow information exchange via multihop crosslinks; that is, if satellites 1 and 2 have a crosslink at the same time as a crosslink for 2 and 3, then 1 is able to crosslink with 3. The algorithms do not incorporate any downlink contention handling, when multiple satellites attempt to talk to the same ground station at the same time. Not incorporating contention handling does not have a significant effect on the simulation results because the number of satellites in a constellation is spaced such that overlapping downlinks rarely, if ever, occur (e.g., only about six overlapping downlinks between satellites were found in a 12 h period for the stitched Walker constellation, over about 300 total downlinks). This assumption may be more important for constellations with larger numbers of satellites, in which case some kind of frequency division multiple access scheme could be used or contention in downlinks could be purposefully scheduled out, as demonstrated by Castaing [35].

We assumed that commlinks to a backbone communications constellation are only available at certain, fixed intervals for every satellite. Also, no gain patterns were incorporated for downlink, crosslink, and commlink antennas; we assume that the satellite either can hold the right attitude (downlink) or has an isotropic antenna (crosslink, commlink). Though a truly isotropic antenna is not achievable, similar performance could be achieved with two complementary half-wave dipole antennas (each with a torus-shaped gain pattern, maximum gain of 2.15 dBi, and half-power beam width of 90 deg [36]). We conservatively assume very low data rates sufficient for planning information sharing over crosslink and commlink (9600 bps).

Battery recharging (ES replenishment) only occurs during recharge mode. This assumption misses some of the potential benefit of the spacecraft serendipitously recharging during other activities but is conservative in nature. Also we assume that a slew must always immediately precede observations, recharges, and downlinks but does not have to precede crosslink and commlinks (due to the isotropic assumption).

In terms of the constellation's orbital geometry, we make several assumptions. The satellites are unable to modify their orbits and do not have to account for collisions between satellites. We also assume that orbits can be determined well enough that satellite orbit position uncertainty has a negligible effect on onboard planning quality over a 24 h period (which is supported by uncertainties found in the literature [37–40]). The satellites' attributes are summarized in Table 3. For this reason, we can assume accurate information about activity windows is available to the satellites in advance of their onboard planning, uplinked by a ground station at regular intervals.

E. Performance Metrics

Several metrics were used for assessing the coordinated performance of the constellation. The first is the average revisit time for a given region and sensor combination, which gives a good high-level assessment of how well the satellites cooperate. The average revisit time for a given region r and sensor s is calculated by averaging all the time differences between the end of one observation of that region by any satellite and the start of the next by any satellite (as well as the start and end time of the overall simulation). This is shown in Eq. (1). The average revisit time for r and s is $\bar{t}_{r,s}$; the set of observations for r, s is $\text{Obs}_{r,s}$ (with cardinality n); the start and end times of each observation i are $t^S_{a,i}$ and $t^E_{a,i}$ (respectively); and t_{start} and t_{end} are the start and end times of the simulation (the symbol a signifies "activity", of which "observation" is an instance). These average times were subsequently averaged across all regions for a given simulation case (producing the results in Figs. 8–10). The algorithms were designed with the objective to minimize these average revisit times for each region:

$$\bar{t}_{r,s}(\text{Obs}_{r,s}) = [(t_{\text{end}} - t^E_{a,n}) + (t^S_{a,n} - t^E_{a,n-1}) + \ldots + (t^S_{a,1} - t_{\text{start}})]/(n+1) = \left[t_{\text{end}} + \sum_{i \in \text{Obs}_{r,s}} (t^S_{a,i} - t^E_{a,i}) - t_{\text{start}}\right]/(n+1) \tag{1}$$

The second set of metrics is the number of information-sharing communications links available and executed by the constellation. This metric serves as an initial assessment of how well a given constellation should perform at information sharing.

The final set of metrics assesses the information-sharing performance of the constellation. The first of these metrics, constellation latency (CL), measures how long it takes for information to propagate between pairs of satellites in the constellation. For a given satellite pair, CL is the average

time between the first scheduling of an observation activity by the originating ("from") satellite ($t_{\mathrm{FS},i}$) to the first reception of information about this scheduled observation by the receiver ("to") satellite ($t_{\mathrm{FR},i}$). It is averaged over all observations $\mathrm{Obs}^{k,l}$ received by the receiver satellite l from the originating satellite k. The second of these metrics, constellation initial timeliness (CIT), measures how timely the shared planning information was for the receiving satellite. For a given satellite pair, CIT is the average time between the first reception time of an observation of the originating ("from") satellite by the receiver ("to") satellite and the start of the next matching observation of the receiver satellite ($t^S_{M,i}$). A "matching" observation is one that has a start time within a fixed time cutoff before the start of an observation of the same region by the receiving satellite. Using this cutoff forces the metric to only look at observations that are timely and relevant for planning purposes. If no matching observations are found at all, the metric simply does not have a value for that direction, as shown in Figs. 14, 15. CIT is averaged over all matching observations received. CL and CIT are subsequently averaged over all from–to satellite pairs to produce the results in Figs. 12, 13. A low CL and a high CIT are desired, meaning that it takes a small amount of time for information to propagate across the constellation and that information is very timely when it arrives. In their current version, the algorithms do not explicitly aim to reduce CL and increase CIT; however, these metrics help to understand the performance of the constellation as a whole. These metrics are calculated by Eqs. (2) and (3), where the cardinality of $\mathrm{Obs}^{k,l}$ is m:

$$\overline{\mathrm{CL}} = \left[\sum_{i \in \mathrm{Obs}^{k,l}} (t_{\mathrm{FR},i} - t_{\mathrm{FS},i}) \right] / m \tag{2}$$

$$\overline{\mathrm{CIT}} = \left[\sum_{i \in \mathrm{Obs}^{k,l}} (t^S_{M,i} - t_{\mathrm{FR},i}) \right] / m \tag{3}$$

11

III. Resource-Aware SmallSat Planner and Limited Communication Constellation Coordinator Algorithms

The operations automation problem was broken down into two phases: 1) coordination of observations at the constellation level, and 2) planning of onboard activities on individual spacecraft. At the constellation level, satellites share information about planned observations via crosslinks, downlinks, and commlinks to a communications backbone constellation. Each satellite uses its latest knowledge about other satellites' plans to determine preference weightings for its own possible observation activities. Using these preferences, the satellite performs a lower-level planning process during which it selects an achievable set of observations and crosslinks while keeping its own onboard resource constraints in check.

The following sections detail the satellite operational model used for the planning process, the low-level RASP planner, and the constellation-level LCCC algorithm.

A. Resource-Aware SmallSat Planner

The RASP algorithm was developed to autonomously plan and schedule activities onboard a resource-constrained small satellite. We briefly describe the RASP algorithm here; additional detail can be found in [29–31]. The algorithm has some similarities with the ASPEN/CASPER algorithms developed at NASA JPL [18] in that it evaluates the feasibility of performing activities based on onboard resource usage, but it 1) uses a simpler model focused specifically on a resource-constrained satellite, and 2) constructs an entire activity sequence onboard the satellite, as opposed to modifying a sequence uploaded from the ground.

Activity planning constitutes the selection of a set of activities (an "activity sequence") from the operational state machine (Fig. 1) that allows the satellite to execute the highest weighted observation activities and as many crosslink activities as possible while maintaining onboard resources within constraints. Scheduling is the assignment of a set of start and end times to every activity in the plan (an "activity timeline") such that an overall score function is maximized as well as the determination of acceptable trajectories for onboard resource states. The RASP algorithm finds a suboptimal but constraint-consistent activity timeline within a given planning horizon (t_h) given a set of initial observation and crosslink windows.

The following subsections describe the main components of RASP: the inputs to the algorithm, the mixed-integer linear program (MILP) formulation used to schedule an activity timeline, and the search process used to find a feasible activity sequence.

1. *Resource-Aware SmallSat Planner Inputs*

A simulation of the satellites' orbits is run to derive observation, crosslink, recharge, and downlink windows. Observation activity windows correspond to the times when the subsatellite point (the intersection of the Earth's surface with the vector from the Earth's center to the satellite) is within a target region of latitude and longitude ranges (which we call an "observation region"). Observation regions are nonoverlapping. Crosslink ("Xlnk") windows correspond to the times that the satellite is within 2400 km of another satellite. Downlink ("Dlnk") windows occur whenever the satellite is above a fixed elevation mask (10 deg) as viewed by the ground station. Recharge ("Rech") windows occur whenever the satellite is illuminated by the sun.

Given the time windows over a specified time horizon t_h, RASP constructs an initial activity sequence by assuming a single observation or crosslink activity occurs during each of their respective windows, with idles and the required slews in between. A notional initial sequence is shown in Fig. 4. Observation activities are weighted based on their importance for coordinated performance across the constellation, as detailed in Sec. III.B.2. If an observation and crosslink overlap, preference is given to the observation. In the current version of RASP, no special consideration is given to the importance of crosslinks, which could negatively impact information sharing performance. The planner simply tries to schedule all crosslinks in this initial study. In future work, we plan to explicitly consider the utility of each observation and communications activity.

2. *Activity Timeline Optimization*

Given an activity sequence, the scheduler component of RASP attempts to find an activity timeline that maximizes Eq. (4). An activity timeline consists of an ordered list of time points $t^S_{a,i}$ and $t^E_{a,i}$ where $i \in [1, N]$, which represent the start and end times of each activity, respectively. N is the number of activities. The symbol a signifies a high-level activity, such that $a \in \mathrm{Act} \triangleq \mathrm{Obs} \cup \mathrm{Xlnk} \cup \mathrm{Dlnk} \cup \mathrm{Rech} \cup \mathrm{Slew} \cup \mathrm{Idle}$, where each set in the overall union contains all the observation, crosslink, downlink, recharge, slew, and idle activities, respectively (for the given satellite in the given planning horizon). This optimization is formulated as a mixed-integer linear program (MILP):

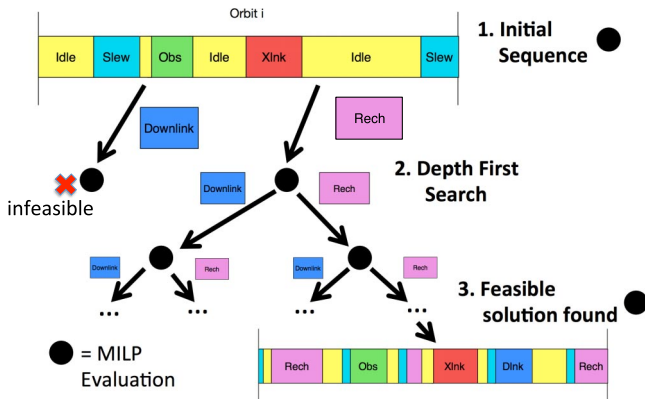

Fig. 4 Illustration of depth-first search process used in RASP to find a feasible activity timeline. The MILP scheduling optimization is solved at each new activity sequence (large black dots). The search attempts to add as many activities in a row as possible to find a solution quickly. If an infeasible path is taken, the search will simply continue from a preceding level. Ellipses indicate a continuation of the search tree.

$$\max\left[\sum_{i\in\text{Obs}} w_o'(t^E_{a,i}-t^S_{a,i}) - w_d\sum_{i\in\text{Dlnk}}\dot{d}_i(t^E_{a,i}-t^S_{a,i}) + w_e\sum_{i=1}^{N}\sum_{j=1}^{i}\dot{e}_j(t^E_{a,j}-t^S_{a,j})\right] \tag{4}$$

subject to

$$t^S_{a,1}=0, \qquad t^S_{a,i}\le t^E_{a,i}, \qquad t^E_{a,N}=t_h \quad \forall\ 1\le i\le N \tag{5}$$

$$t^E_{a,i}=t^S_{a,j} \quad \forall\ i\ge 1, \qquad j\le N | j=i+1 \tag{6}$$

$$t^S_{a,i}\ge t^{S,W}_{a,i}, \qquad t^E_{a,i}\le t^{E,W}_{a,i} \quad \forall\ i\in \text{Obs}\cup\text{Dlnk}\cup\text{Rech} \tag{7}$$

$$t^S_{a,i}=t^{S,W}_{a,i}, \qquad t^E_{a,i}=t^{E,W}_{a,i} \quad \forall\ i\in \text{Xlnk} \tag{8}$$

$$t^E_{a,i}-t^S_{a,i}\ge \text{MinDur}_i \quad \forall\ i\in\text{Act} \tag{9}$$

and

$$\begin{aligned}
&\text{RS}_{\text{init}}+\dot{r}_1(t^E_{a,1}-t^S_{a,1})\le \text{UB}_{\text{RS}}+M(1-z^{\text{UB}}_{\text{RS},1}),\\
&\text{RS}_{\text{init}}+\sum_{i=1}^{2}[\dot{r}_i(t^E_{a,i}-t^S_{a,i})]\le \text{UB}_{\text{RS}}+M(1-z^{\text{UB}}_{\text{RS},2}),\\
&\dots\\
&\text{RS}_{\text{init}}+\sum_{i=1}^{N}[\dot{r}_i(t^E_{a,i}-t^S_{a,i})]\le \text{UB}_{\text{RS}}+M(1-z^{\text{UB}}_{\text{RS},N})
\end{aligned} \tag{10}$$

$$\begin{aligned}
&\text{RS}_{\text{init}}+\dot{r}_1(t^E_{a,1}-t^S_{a,1})\ge \text{LB}_{\text{RS}}-M(1-z^{\text{LB}}_{\text{RS},1}),\\
&\text{RS}_{\text{init}}+\sum_{i=1}^{2}[\dot{r}_i(t^E_{a,i}-t^S_{a,i})]\ge \text{LB}_{\text{RS}}-M(1-z^{\text{LB}}_{\text{RS},2}),\\
&\dots\\
&\text{RS}_{\text{init}}+\sum_{i=1}^{N}[\dot{r}_i(t^E_{a,i}-t^S_{a,i})]\ge \text{LB}_{\text{RS}}-M(1-z^{\text{LB}}_{\text{RS},N})
\end{aligned} \tag{11}$$

The score function in Eq. (4) attempts to maximize three items: the weighted sum of all observation durations in the activity timeline (summation 1 from left-hand side, in minutes), the total amount of data downlinked over the activity timeline (summation 2, in bytes), and the average ES state over the course of the activity timeline (double summation, in watt hours). The weighting factors on these terms allow

the summation of different units. The value w'_o is a normalized version of the weighting given to each specific observation by LCCC, w_o, as explained in Sec. III.B.2 and Algorithm 2. It is normalized by the total potential observation time summed up across all observation window lengths.

The $\dot{d}_i$ and $\dot{e}_j$ terms correspond to the DS usage rate and ES production rate for activities i and j, respectively (limits are enforced on ES and DS, as detailed in Sec. IV.C and Table 3). For the ES double summation, the outer summation ($i = 1$ to N) accounts for the ES state at the end of all activities in the activity sequence, and the inner summation propagates the ES state forward through the activity timeline by accounting for ES changes over all activities j up to activity i. For example, if $i = 3$, then the inner loop sums the energy added or subtracted over activities 1, 2, and 3: $\dot{e}_1(t^E_{a,1} - t^S_{a,1}) + \dot{e}_2(t^E_{a,2} - t^S_{a,2}) + \dot{e}_3(t^E_{a,3} - t^S_{a,3})$. If a constant is added to this sum to represent the energy at the current time, we have the energy state at the end of activity 3 (though we do not need to include this constant term because it is the same for all possible timelines in the current planning horizon). The weighting terms w_d and w_e are calculated as

$$w_d = u_{\mathrm{DS}}/(\mathrm{UB}_{\mathrm{DS}} - \mathrm{LB}_{\mathrm{DS}}) \tag{12}$$

$$w_e = [u_{\mathrm{ES}}/(\mathrm{UB}_{\mathrm{ES}} - \mathrm{LB}_{\mathrm{ES}})]/N \tag{13}$$

Equation (12) expresses that the total amount of data downlinked over an activity timeline is normalized by the range between DS bounds (where $\mathrm{UB}_{\mathrm{DS}}$ and $\mathrm{LB}_{\mathrm{DS}}$ represent the upper and lower bounds, respectively) and multiplied by a unitless "urgency factor" u_{DS}, which effectively tunes the algorithm's preference for downlinking data. If this factor is set to 0, RASP will not care at all about downlinking data outside of its necessity to keep DS within bounds [Eqs. (10) and (11) actually force DS and ES to stay within bounds]. Equation (13) is a similar expression, except that the additional normalization by the number of activities, N, means that the algorithm minimizes average ES state. These urgency factors were fixed at 1 based on the analysis in [29], which delivers a good balance between resource management and observation time maximization.

Equation (5) enforces a planning window from 0 to t_h and ensures that the end of every activity follows its start. Equation (6) forces activity j to immediately follow activity i ($j = i + 1$). Equation (7) forces the Obs, Dlnk, and Rech activities to fall within their windows; and $t^{S,W}_a$ and $t^{E,W}_a$ signify the start and end of the relevant time window. Equation (8) forces Xlnk activities to start and end exactly on their window bounds, which ensures different satellites commit to crosslink at the same time. Equation (9) enforces activity minimum durations. The N equations in Eqs. (10) and (11) enforce resource constraint upper bounds (UB) and lower bounds (LB), respectively; the RS signifies that these equations hold for both resource types: ES and DS. We use the "big M" method to select whether specific constraints will or will not be enforced [41]; hence, M is a large integer, and $z^A_B \in \{0, 1\}$ is a variable that decides whether the constraint for the given activity number is enforced. Constraint enforcement is covered more in the next subsection.

3. *Activity Sequence Construction Through Greedy Search*

RASP uses the selective enforcement of constraints in Eqs. (10) and (11) as a mechanism for determining where to add downlink and recharge activities to arrive at a final plan with all constraints enforced. Detailed coverage of the RASP algorithm is out of the scope of this paper; the reader is referred to [29] for full coverage of the algorithm's details. The algorithm performs a depth-first search through a tree of modified activity sequences constructed from the initial activity sequence, solving the MILP optimization each time. Children activity sequences are created by replacing an idle activity with a resource management activity, an activity of type Dlnk or Rech, one at a time to the parent activity sequence. More Slew and Idle activities are added as necessary to conform to the operational state machine. This process of search through incremental activity sequence modifications is shown in Fig. 4.

Adding these activities allows the algorithm to progressively enforce more of the driving constraints (DSUB and ESLB), pushing toward the goal state of having all constraints enforced. When a new activity is added, the algorithm attempts to solve the MILP with the appropriate resource constraint set enforced up to the location where the activity was added. A heuristic function is used to push the algorithm to progressively enforce more constraints, while also trying to increase the score for the timeline.

When a timeline is found that satisfies all the constraints in the MILP, it is returned. This search process is limited to a timeout period, after which the input sequence is considered to have failed, and a reduction is made to it (by removing an observation or crosslink) and RASP is run again. The timeout period was set to 7 s, which was found to sufficiently balance the RASP optimization process for feasible initial activity sequences with cutting off the search for infeasible sequences. This repeats until RASP finds a fully constraint-satisfying timeline, even if it has no observations or crosslinks. The RASP algorithm as currently implemented is nonoptimal and noncomplete.

B. Limited Communication Constellation Coordinator

The LCCC algorithm calculates weightings for observation activities from planning information obtained via a "weak" form of distributed consensus. The weightings are selected to direct RASP to minimize the average revisit times for all regions being observed, over all three sensor types. The following subsections explain the consensus mechanism used, and how weightings are calculated.

1. *Weak Consensus over Shared Planning Information*

The weak distributed consensus mechanism works such that each satellite maintains its own database of the most up-to-date observation planning information obtained from all satellites and updates specific entries in this database whenever new information becomes available through crosslinks, downlinks, or commlinks. The mechanism is described as weak because not all the satellites in the constellation are required to come to a consensus on the information stored in each of their separate databases. Instead, consensus is only achieved between the satellites (or satellite and ground station) involved in a communications link. The satellites in the crosslink may receive secondhand planning information from other satellites that were participants in previous crosslinks. This secondhand information could be stale and even mislead the satellites to choose conflicting sensors for their observations. Other "strong" consensus-based task assignment algorithms such as the Consensus-Based Auction Algorithm (CBAA) detailed by Choi et al. [28] are designed to eliminate conflicting assignments between agents through many iterations of assignment and communication across the full network. CBAA is very capable in a scenario where interagent communication transactions occur frequently, relative to the planning horizon. The relatively infrequent availability of communications links in our scenario necessitates an approach with less stringent requirements on consensus. The LCCC algorithm essentially eases the communication requirements on the satellites by accepting the possibility of conflicting observations if the satellites change plans after initially sharing them.

Algorithm 1 Update satellite *i*'s knowledge of planned observations for satellite *j*

1: **procedure** UPDATEOBSERVATIONS $(\boldsymbol{o}_i, \boldsymbol{o}_j)$
2: **for** $theirObs \in \boldsymbol{o}_j$ **do**
3: $myObs \leftarrow$; FINDMATCHING $(theirObs, \boldsymbol{o}_i)$
4: **if** $myObs \neq \emptyset$ **then**
5: $theirLastUpdateTime \leftarrow$; LASTUPDATED $(theirObs)$
6: $myLastUpdateTime \leftarrow$; LASTUPDATED $(myObs)$
7: **if** $myLastUpdateTime < theirLastUpdateTime$ **then**
8: $myObs \leftarrow theirObs$
9: **end if**
10: **else**
11: $o_i \leftarrow \boldsymbol{o}_i \cup theirObs$
12: **end if**
13: **end for**
14: **end procedure**

Every satellite, or "agent", maintains a list $\boldsymbol{o}_i$ of the observations both planned and already executed for every satellite in the constellation, including itself. All observation activities across the constellation are uniquely identified by creation time and target region. Whenever a satellite settles on an activity sequence (up to its current time plus time horizon t_h), it adds its list of planned observations to $\boldsymbol{o}_i$. If an observation was already scheduled during a previous planning period, its time parameters are simply updated. All updated or added observations are tagged with an appropriate update time. During crosslink activities, satellites trade their lists of planned observations between each other and update any observations in their own list that are out of date (i.e., the update time on their version of the observation is earlier than the update time provided by another satellite).

This process is reflected in Algorithm 1. Agent i receives list $\boldsymbol{o}_j$ from all other agents j and uses these to update its own list. The FINDMATCHING() procedure on Line 4 returns a matching observation if it exists, or returns the empty set if satellite i has not heard about this observation yet. Line 8 tests if the other agent's matching observation is more up-to-date, and updates the agent's observation (Line 9) if it is. Line 12 stores the other agent's observation if it is completely unknown.

One important assumption in this model is that all satellites have access to the same global clock; this is achievable with GPS time for low-Earth-orbit constellations. Another important requirement for this method to work well is a high degree of dynamic network connectivity. That is, as the satellites orbit and they perform opportunistic crosslinks and downlinks, we desire a given agent to hear from all other agents that could potentially affect its weighting for a particular observation.

2. Selecting Observation Weightings from Shared Planning Information

A given observation's weighting is based on what the agent knows about all agents' planned observations. An observation is weighted more heavily the farther it is away from the closest preceding observation of the same region, with the same sensor. Algorithm 2 determines, for every possible observation from the satellite's current time to time horizon t_h, the closest preceding observation for every sensor type. It assigns a weight for performing the observation with each sensor type and selects the highest weighting over all sensors. The highest weighting is RASP's w_o input for that observation. This weighting algorithm is called by RASP every time it tries to schedule a set of observation and crosslink windows (i.e., solves the MILP in Sec. III.A.2). RASP updates both the observation timings and selected sensor types after successful scheduling.

Algorithm 2 works in the following way. A list of observation windows for satellite i from the current planning window (current time to t_h), $\boldsymbol{o}^W_{i,i}$, is provided along with $\boldsymbol{o}_i$ and the start and end times for the weighting window $t^S_{\text{weighting}}$ and $t^E_{\text{weighting}}$. Note that the i, i subscript indicates that, from the lists of observation windows for all satellites ($\boldsymbol{o}^W_i$), the list for satellite i is selected ($\boldsymbol{o}^W_{i,i}$). List $\boldsymbol{o}^W_{i,i}$ is assumed to be sorted in ascending order by start time. A list of weights for all observation windows is instantiated on line 2. Any observation windows previously scheduled from the current planning window are removed from $\boldsymbol{o}_i$ on line 3. This allows the windows in $\boldsymbol{o}^W_{i,i}$ to be added back in on line 13 without any fear of duplication, and so they can also be considered in the weightings. Each observation window o^W is looped through in increasing start time (line 4). A list of sensor (s) specific weights, $weights_W$, for the window is instantiated on LINE 5. All sensors are looped through and added to this list in lines 6 to 11. The FINDNEARESTPRECEDING() procedure finds the start of the nearest preceding observation activity or window matching o^W's region from $\boldsymbol{o}'_i$, for the specified sensor. If no preceding observation is found after time $t^S_{\text{weighting}}$, then that time is returned. A time difference is calculated from the preceding time and the start of o^W (line 8) and normalized by the length of the weighting window (line 9) to give a weight for the current sensor.

Algorithm 2 Calculate the weighting of all observation windows within current planning horizon

1: **procedure** CALCULATEWEIGHTINGS $(\boldsymbol{o}^W_{i,i}, \boldsymbol{o}_i, t^S_{\text{weighting}}, t^E_{\text{weighting}})$
2: $\boldsymbol{w}_o \leftarrow \emptyset$
3: $\boldsymbol{o}'_i \leftarrow \boldsymbol{o}_i \setminus \boldsymbol{o}^W_{i,i}$
4: **for** $o^W \in \boldsymbol{o}^W_{i,i}$ **do**
5: $weights_W \leftarrow \emptyset$
6: **for** $s \in \{A, B, C\}$ **then**
7: $t_{\text{prec}} \leftarrow$; FINDNEARESTPRECEDING $(o^W, \boldsymbol{o}'_i, t^S_{\text{weighting}}, s)$
8: $\Delta t \leftarrow$; START $(o^W) - t_{\text{prec}}$
9: $weight_W \leftarrow \Delta t / (t^E_{\text{weighting}} - t^S_{\text{weighting}})$
10: $weights_W \leftarrow weights_W \cup weight_W$
11: **end for**
12: $\boldsymbol{w}_o \leftarrow \boldsymbol{w}_o \cup$ MAX $(weights_W)$
13: $\boldsymbol{o}'_i \leftarrow \boldsymbol{o}'_i \cup o^W$
14: **end for**
15: **return** $\boldsymbol{w}_o$
16: **end procedure**

After all sensor-specific weights are determined, the sensor corresponding to the maximum weight for the observation is selected on line 12. If all the weights are the same, sensor A is chosen.

An important feature of this weighting algorithm is that it has an explicit preference for earlier observations. This means that, if two satellites have two observations with start times that are separated by a small temporal distance (say on the order of seconds), then given that both satellites know about the two observations, the earlier observation will be weighted much higher than the later one. This is because the second observation sees the first as its "nearest preceeding", and its calculated weight is much lower. This degree of objectivity encourages satellites to commit early to observations and stick with their decisions across multiple RASP planning horizons, which is important for ensuring that shared planning information is timely and relevant. Also note that all physically possible observation windows in $\boldsymbol{o}_{i,t}^{W}$ are considered in the weighting calculation, even those which may be operationally impossible to execute. This has a negligible effect, though, because weightings are recalculated again at the next planning time.

IV. Constellation Simulation Details

A simulation was constructed and executed for a coordinated constellation using the RASP and LCCC algorithms. The following subsections give details on the choice of parameters for the simulation. Two very important considerations were the choice of orbits for all the satellites in the constellation (the first subsection) and the use of communications links for information sharing (the second subsection). The third and fourth subsections give specifics on the activity and resource parameters for the satellites. The fifth subsection talks about the software developed for the simulation.

A. Constellation Orbital Geometry Parameters

The selection of orbits for the CubeSats in the constellation is important to achieving both good geometric coverage of the Earth's surface and providing sufficient communications connectivity. We investigated three constellation orbit architectures: 1) a "plain" Walker star, 2) a "stitched" Walker star, and 3) an ad hoc constellation.

The plain Walker constellation is based on the original Walker star architecture introduced by Walker [42,43]. The architecture has been extensively studied in the literature in terms of revisit time and coverage metrics [44–49]. The Walker star spreads its satellites over multiple, regularly spaced, polar orbits, which provides good coverage over most of the Earth's surface. We also included a modified Walker constellation, which we refer to as a "stitched" Walker star. It modifies the basic architecture by adding two additional lower inclination orbits that "stitch together" the star orbits. That is, the satellites in these orbits are placed in true anomaly such that they are able to perform frequent crosslinks with satellites in the star orbits, allowing more information exchange across the constellation as a whole.

A second architecture was considered that reflects an easier to field case. We refer to this architecture as "ad hoc". It is formed opportunistically from CubeSat deployments over multiple launches. For this work, it is based on the second of the two ad hoc constellations analyzed by Marinan et al. from an assessment of launch opportunities for CubeSats in the 2013 calendar year [45]. The ad hoc represents a case that is more likely to be achieved in the near future given the high costs of launch and the usual placement of CubeSats as "secondary payloads". Yet it is useful to compare with the Walker constellations because these are based on such a commonly studied geometry.

The parameters for the geometries are summarized in Table 1 and detailed in the Appendix. Each architecture has 18 satellites spread over six orbits, and the three satellites within a single orbit are assumed to remain at a 120 deg true anomaly separation over the whole simulation. These numbers are simply for convenience; three satellites per orbit provides good coverage in that orbit without needing to simulate too many satellites, and six orbits provides diversity in coverage between orbits. The plain Walker is in an i: $T/P/F = 90$: $18/6/0.5$ configuration, and the stitched Walker includes a main 12-satellite component in an i: $T/P/F = 90$: $12/4/0.33$ configuration (where T is total number of satellites in the pattern, P is the number of planes, and F is the relative spacing number).

B. Constellation Simulation Cases

A given constellation geometry is simulated in seven cases with different planning and communications contexts, as summarized in Table 2. The planning context reflects how observation activities are weighted. "Onboard" (or online) refers to the full use of the LCCC algorithm to weight observations based on planning information from the constellation and then activity scheduling with RASP. In the "advance weighting" context, observations are weighted across all satellites at the beginning of the simulation using a simple greedy algorithm. The greedy algorithm looks at all

Table 1 Summary of constellation orbital parameters

Constellation geometry type	Number of planes	Altitude, km	Satellites per plane	Inclination, deg
Plain Walker	6	600	3	90
Stitched Walker	6	500, 600	3	90, 56
Ad hoc	6	600 to 825	3	98, 51, 52

Table 2 Summary of constellation planning and communications contexts

Case	Planning context	Communications context	Description
1	Advance weighting	No info sharing	All observation weights calculated at simulation start; no sharing of planning info
2	Random selection	No info sharing	Observation weights randomly calculated; no sharing of planning info
3	Onboard	No info sharing	Observation weights calculated by LCCC; no sharing of planning info
4	Onboard	Downlink	Observation weights calculated by LCCC; planning info shared via downlink with single ground database
5	Onboard	Crosslink	Observation weights calculated by LCCC; planning info shared via crosslink with separate databases on each satellite
6	Onboard	Crosslink + downlink	Observation weights calculated by LCCC; planning info shared via both downlink and crosslink
7	Onboard	Commlink + downlink	Observation weights calculated by LCCC; satellites routinely share info with an external constellation and ground

the satellites' observations for each region and progresses through them in increasing start time, assigning to each observation in turn the sensor that was least recently used (starting at the beginning with A) and a weight in the same manner as line 9 in Algorithm 2. We assume all satellites begin the simulation with full knowledge of these weightings, in this case. In the "random selection" planning context, the sensor is simply randomly selected when each observation is planned by RASP, and its weighting is calculated at that time.

The communications context reflects the satellites' use of communications links to share planning information between each other. Each satellite maintains its own database of the planning information for every other satellite in the constellation, which it updates when it receives more recent information. "No info sharing" models a noncooperative constellation, where satellites do not share information (though non-info-sharing data downlinks are still performed). In "downlink", each satellite synchronizes its database with a ground-based database whenever it performs a downlink. All ground stations are assumed to be linked to a single database. In "crosslink", the satellites only share planning information via crosslinks, which can be performed whenever satellites approach within 2400 km of each other. All satellites involved in a single crosslink synchronize their databases with each other. Multihop crosslinks are assumed possible, with a single satellite able to function as a bridge between multiple others. "Crosslink + downlink" features both sharing methods. "Commlink + downlink" uses regularly spaced commlinks instead of crosslinks. The commlink windows are spaced at 20 min intervals, at the same times across all satellites. This number was chosen to be the same as the replanning interval, specified in Sec. IV.E, and so the satellites share once every time after replanning. This simple commlink model was adopted because it provides a sufficient update frequency and works for the current version of RASP, which cannot model commlink utility. The commercial communications constellation is connected to the same database as the ground stations.

C. Satellite Attributes and Resource Constraints

We consider a set of CubeSats that are similar in performance to the Micro-Sized Microwave Atmospheric Satellite (MicroMAS) and Microwave Radiometer Technology Acceleration (MiRaTA) CubeSat science missions [29,50–52]. These CubeSats both perform remote sensing observations of the Earth's atmosphere and are representative of the increasingly complex missions that can be performed with small satellites. Both spacecraft have a set of subsystems necessary for achieving their mission objectives, including systems for electrical power distribution and storage, communications, command and data handling, attitude determination and control (using a suite of sensors and actuators, but no propulsion), and a scientific payload with one or more instruments.

The satellites' characteristics are summarized in Table 3. For this work, we model all the satellites in the constellation with 20 W · h of energy storage and a goal to minimize the amount of science data stored onboard. Note that the data storage minimization is a proxy for reducing data latency in the current version of RASP and LCCC. Energy is consumed at different rates depending on the spacecraft mode, but it can only be produced (at 24.8 W) during a dedicated recharge mode when the satellite points its solar panels directly at an optimal angle to the sun. Science data is only produced during observations (at the same rate for all sensors), which are restricted to when satellites fly over specific target regions. Onboard data storage is fixed at an upper limit of 100 MB (1 MB $= 1000^2$ B). The satellite model also assumes an active three-axis attitude control system, which performs slews to change the spacecraft's attitude between most activities. The satellites use a downlink data rate of 2.6 Mbps uncoded, based on the Cadet nanosatellite radio [29]. We assume a 30% reduction factor in downlink rate to represent time lost to link maintenance and processing overhead. We assume a fixed elevation mask (10 deg) for downlink availability at all ground stations and that overlapping downlinks from multiple satellites to the same ground station can be handled.

Note that we do not directly consider the energy and time resources required of the onboard processor to run the RASP algorithm. Our RASP timing results show that the algorithm can be run, on a capable computer (2 GHz Intel Core i7 processor, 8 GB of RAM), in under 5 s for a 90 min planning window [29]. This suggests that RASP could feasibly be implemented and run on an embedded computer typical for CubeSat missions. Example embedded computers include the BeagleBone Black, with an ARM Cortex-A8 microprocessor running at 1 GHz with 512 MB of RAM [53], and the Raspberry Pi (Model B) running at 900 MHz with 1 GB of RAM [54]. During a typical run, the RASP algorithm occupied about 350 MB of RAM, running in the relatively RAM-heavy MATLAB environment. An embedded CubeSat computer should achieve comparable performance (roughly half the speed) with RASP, and performance could be boosted even more with an adaptation to a lower-level language such as C. Note that the software is currently single-threaded and could possibly be parallelized for another performance boost, with some adjustments to the RASP search process. These computers do take a relatively large amount of power for a CubeSat (around 2.5 W), but with the short planning time and long planning window, they could be run at a very low duty cycle; operating for 30 s every 20 min would allow RASP to run sufficiently often and provide plenty of time margin for booting the processor and running the algorithm.

As context, we now provide some detail about these parameters from other CubeSat missions. From Klofas' census of CubeSat communications systems, most CubeSats can downlink data rates of 1200 bps up to tens of kilobits per second [5]. Several CubeSats have flown radios operating at much higher data rates, including the Cadet radio from the L3 company (2.6 Mbps) and the radio from the Emhiser company (1 Mbps). The Planet Labs company has launched CubeSats with downlink data rates up to 10 Mbps in S-band [55]. Both MicroMAS and MiRaTA use the L3 Cadet radio. Although the high data rate of the Cadet does require support from a capable ground station, the successful operation of the radio on the DICE mission [56] justifies our use of its parameters in this work. In terms of energy storage, MicroMAS flew a 20 W · h battery, and other missions have flown both less capacity (CanX-2 at 13 W · h) and more (RAX-2 at 30 W · h) [2,3]. For energy production, the number used here (24.8 W) reflects the use of fairly large, double deployed solar arrays on the MiRaTA spacecraft. However, this number is not unrealistic, considering that this only requires about 20 solar cells (producing about 1.3 W each at 1 AU using standard 30% efficiency cells [4]). The data storage amount assumed here, 100 MB, is significantly less than what can be physically stored with standard commercial nonvolatile memory (multiple gigabytes). This limit encourages the RASP algorithm to schedule downlinks to the ground in a timely manner to reduce data latency. This restriction, while somewhat artificial, is necessary for RASP to operate effectively in its current form, and a more realistic model is planned for future work.

Table 3 Summary of satellite attributes and resource constraints

Attribute/resource	Summary
Observation sensors	Three total; sensors A, B, and C
Energy storage	20 W · h maximum, 14 W · h minimum (30% DOD)
Energy production	24.8 W, only in dedicated recharge mode
Energy consumption	Varies by mode; sensors consume 6.7 W each while observing
Data storage	100 MB maximum, 0 MB minimum
Data production	Engineering telemetry: always produced at 10 kbps; sensor payloads: 63 kbps while observing
Data downlink	1.82 Mbps

Table 4 Activity resource production rates and minimum durations

	Operational state						
Parameter	Obs	Xlnk	Clnk	Rech	Dlnk	Slew	Idle
Energy (ES), W	−14.1	−9.3	−9.3	17.0	−16.0	−7.8	−6.4
Data (DS), kbps	73	10	10	10	−1820	10	10
Minimum duration, min	5	1.5	1	1	1	3	0

D. Satellite Activity Resource Usage

The satellites' resource production rates are broken down by activity in Table 4. The final row specifies the minimum required duration of each activity. The acronym "DOD" means depth of discharge of the battery. The rates in Table 4 are generic and based loosely on the MiRaTA spacecraft [29] as well as the parameters in Table 3. We assume that the sensor payload produces data (at 63 kbps) during the entire observation ("Obs") activity. Crosslink ("Xlnk") energy usage is based on a radio with 10 W input power and a transmit duty cycle of 33% during an actual crosslink. The minimum duration times are considered to be representative of CubeSat operations. The slew duration is long to be conservative. The crosslink duration was based on data from the constellation simulation; it was set long enough to allow multihop crosslink events to happen often. Note that, with the 2400 km crosslink limit, the physical windows for these activities were generally much longer than 1.5 min. The Commlink ("Clnk") activity duration was chosen to allow enough time for establishing a link. Recall that, in this work, crosslinks and downlinks are not used to transfer data (this is an item for future work). Thus, the higher the data production rate by the payload, the more data needs to be downloaded in downlink, which could reduce the number of observations and other activities performed. From simulation results, we found that the overall constellation performance did not change significantly depending on the data production rate because downlink windows were generally not fully used in the nominal case.

E. Simulation Software Environment

Several software components were used or developed as part of this process. The flow of information between the components is depicted in Fig. 5. Orbital geometry was analyzed using the Systems Tool Kit (STK) software package from AGI and used to determine time windows for observation, crosslink, downlink, and recharge activities for each satellite.

The RASP algorithm was implemented in the MATLAB language from MathWorks, using the linprog() function and dual-simplex optimizer for linear program solution [57]. RASP was run in a receding horizon fashion; the algorithm was used to plan an activity timeline over a certain planning horizon, the satellite's state was propagated forward for a period of time using that timeline, and then activities were replanned. We developed a software package in Python to simulate the on-orbit operation of a constellation with an arbitrary number of satellites planning and scheduling though RASP and LCCC. The simulation initially ingests orbit data from the STK analysis. Individual satellite information is stored in a custom Python class. The simulation keeps track of a global clock for the constellation, propagates satellite resource states forward, calls the RASP algorithm for replanning at regular intervals (as well as after crosslinks, commlinks, and downlinks), maintains satellite and ground-planning information databases, and shuttles planning information around as specified by the crosslink, commlink, and downlink activities. The global clock was configured to run with 1 s ticks.

A 24 h simulation was run for each combination of orbit geometry and communications context. That duration includes about 16 orbits for each satellite and includes a large number of observations and crosslinks. We consider this sufficient for this first assessment of the algorithms' performance. The resource bounds and production rates were set per Tables 3, 4. RASP planning was performed over a 90 min planning window ($t_h = 90$), and satellites replanned every 20 min, or after every crosslink, downlink, or commlink in which they obtained updated planning information. Satellite states were propagated forward using the same rates, with a small amount of noise added on top of the rates to simulate model imperfections. The noise was normally distributed about the nominal usage rate, with a standard deviation of 1% of the nominal rate; the noisy rates were allowed to saturate at 95 and 105% of the nominal rate. Seventeen arbitrary observation regions were chosen (detailed in the Appendix). These are indicated along with the nine ground stations in Fig. 3. Note that it took around 10 h to run a simulation case.

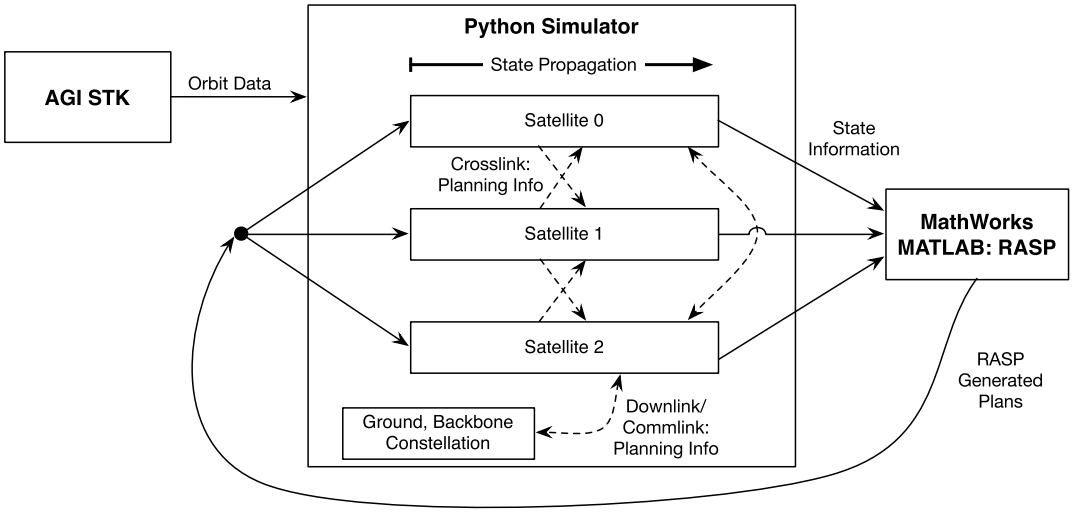

Fig. 5 Software flowchart for the simulation software. Orbit data are generated in STK as text files and imported into a Python script. Several satellite objects (three here) keep track of the satellites' states and call RASP when required. Crosslinks, downlinks, and commlinks share planning information between the satellite objects.

V. Simulation Results

The simulation environment was employed to analyze the performance of the coordination algorithms. Three major analyses were performed on the simulation cases: 1) the effect of constellation orbital geometry and communications context on revisit times, 2) the number of communications links executed over all the cases, and 3) how effectively planning information was shared across the constellations. The following subsections discuss the results from each of these analyses in turn.

A. Revisit Time Performance

The first analysis was performed on the revisit times achieved in each simulation case. As context, Figs. 6, 7 show the available observation windows and executed observations over all sensor types, organized by satellite number, for the stitched Walker constellation in two different simulation cases. Note that only a subset of satellites are included, for clarity. We see that, with no information sharing (Fig. 6), the sensor types are poorly spread out over the 24 h period. Many observations are executed with sensor A up to about 10 h in, then sensors B and C begin to mix in. Much better mixing is seen from the beginning when planning information is shared widely (Fig. 7). In particular, we see a much more diverse selection of sensors for the "R" (Russia) region in the first 5 h of the simulation. We cannot determine from this plot which satellites coordinated most effectively with which other satellites, but we do see that the LCCC algorithm tends to balance out sensor selection across satellites. This balancing of sensor selection tends to balance average revisit times across the constellation. When no planning information is shared, every single satellite simply tries to minimize revisit times for each sensor over all regions by itself.

Figures 8–10 summarize the revisit time results for the three different constellation geometries. The plots summarize the revisit times for each sensor in each simulation case, first averaged over all satellite observations for a particular region and then averaged over all regions. Note that the vertical axes have the same scale across all three plots, and the "no sharing" case extends (off the chart) higher up than shown. Figure 9 supports the trends seen in Figs. 6, 7 for stitched Walker. We see that, with increasing information sharing, progressing from "no sharing" to "Clnk + Dlnk", the average revisit times are much better balanced between the sensors: the "intersensor range" between lowest and highest sensor revisit time

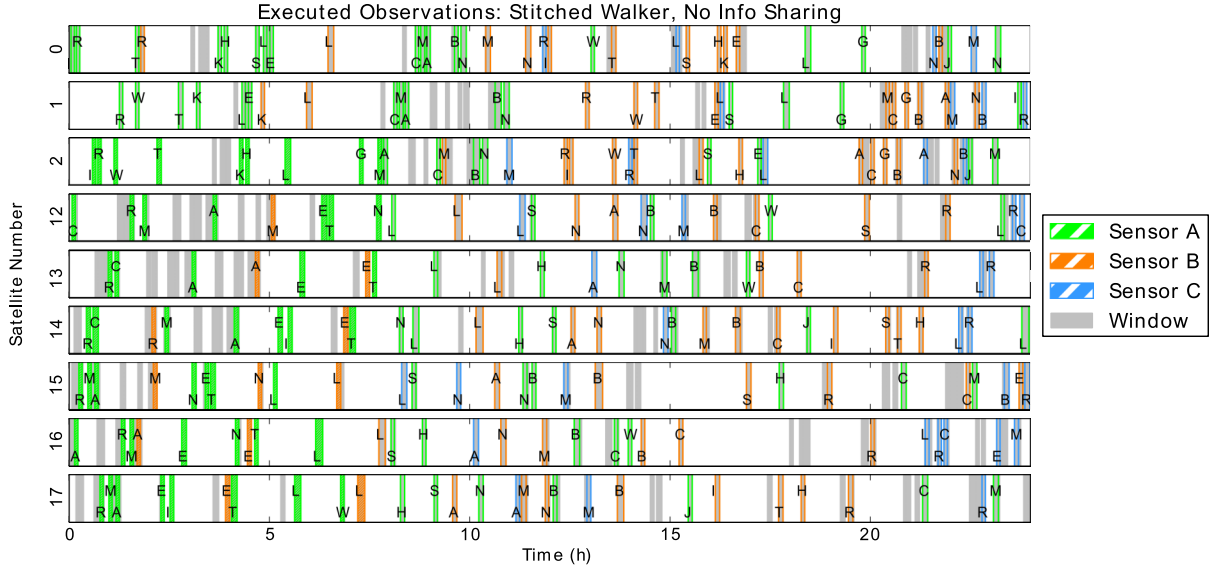

Fig. 6 Executed observations and windows for stitched Walker in no sharing and onboard planning case (case 3). Letters indicate observation region. Note that satellites 3 to 11 are discluded for clarity. Sensor selections are badly balanced across satellites.

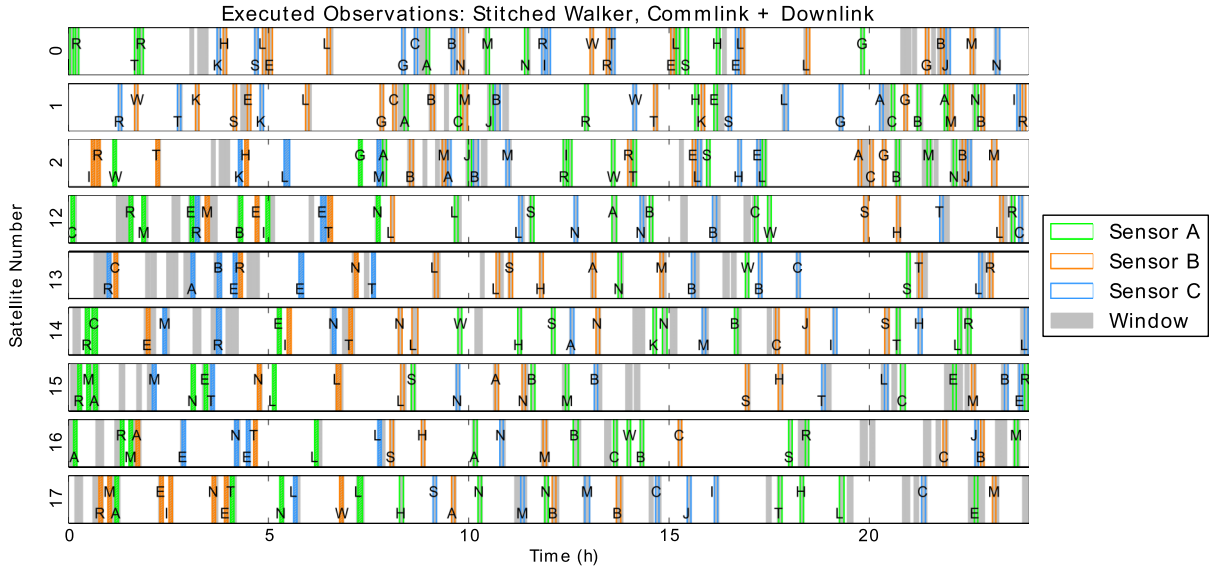

Fig. 7 Executed observations and windows for stitched Walker in commlink + downlink case. Letters indicate observation region. Note that satellites 3 to 11 are discluded for clarity. Sensor selections are balanced much better with increased information sharing.

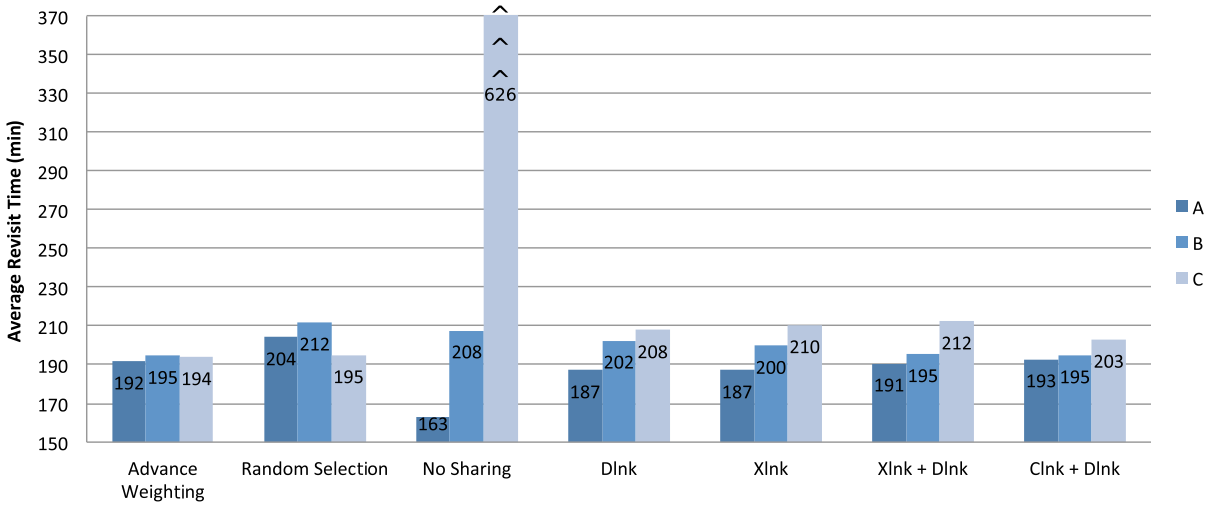

Fig. 8 Average revisit times, averaged over all regions, for the plain Walker constellation.

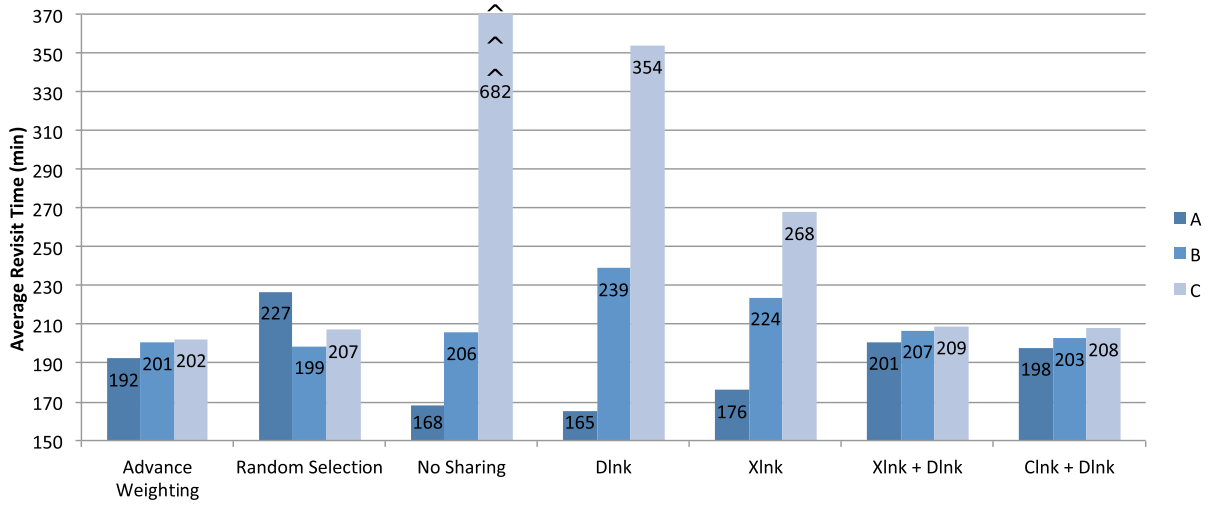

Fig. 9 Average revisit times, averaged over all regions, for the stitched Walker constellation.

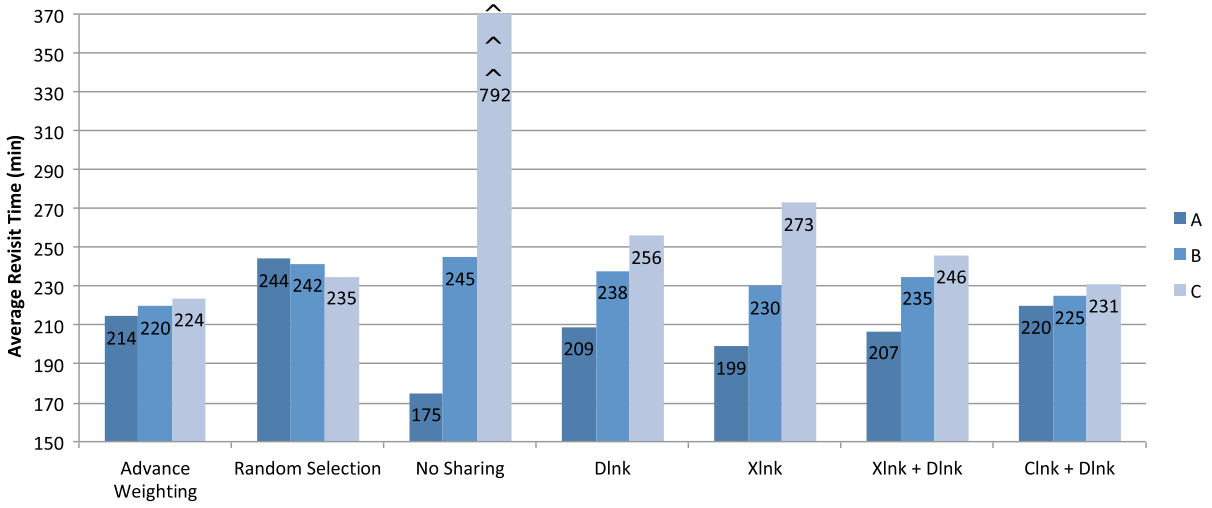

Fig. 10 Average revisit times, averaged over all regions, for the ad hoc constellation.

decreases from 514 to 10 min. Also, if we further average the average revisit time across all three sensors, that average tends to improve (decrease) with more information sharing. The case with Clnk + Dlnk performs the best, as expected due to the satellites' frequent communication with a globally shared planning information database.

But revisit time performance does not change in the same way across all the constellations. We see that plain Walker achieves good performance in all cases, and stitched Walker and ad hoc show a much stronger improvement trend with increasing information sharing. The relatively flat performance for plain Walker suggests that information sharing does not play as strong of a role in this constellation. This conclusion makes sense, considering that the plain Walker constellation was constructed simply for observation coverage, not to maximize

crosslink performance. All constellations settle out to roughly the same intersensor range in the commlink case, about 10 min, though the ad hoc constellation averages about 225 min in average revisit time across all sensors, versus 197 and 203 for the Walker constellations.

The "advance weighting" and "random selection" cases add more insight to the performance of the LCCC algorithm. We see that advance weighting performs slightly better than Clnk + Dlnk in all cases, with the same or lower intersensor range and a lower average across all sensors (194, 198, and 220 min versus 197, 203, and 225 min for plain Walker, stitched Walker, and ad hoc, respectively). This shows that, for the specific application case investigated in this work, online, onboard planning with information sharing performs no better than advance, ground-based planning. But the performance with Clnk + Dlnk is not significantly worse than the advance weighting case (the values are under 3% larger), which shows that there is no significant penalty here for not using advance planning or investing having to perform commlinks. In the random selection case, all intersensor ranges fall within 28 min (compared to 11 min for Clnk + Dlnk), and the averages across all sensors are 204, 211, and 240 min for plain Walker, stitched Walker, and ad hoc, respectively. Although not significantly worse than Clnk + Dlnk (under 7% larger), RASP and LCCC clearly perform better than with no coordination, at least when planning information is shared often through the backbone constellation. In the Xlnk + Dlnk case, the revisit time results are not consistently better than random selection; revisit averages across all sensors are better at 199, 206, and 229 min (plain Walker, stitched Walker, and ad hoc, respectively), but all intersensor ranges fall within 39 min, a larger spread. Performance versus random selection decays significantly for the Xlnk only case. This shows that, with reduced information sharing, the coordination performance is not strictly better than the random selection case for the current application.

B. Communication Links Performance

The second analysis conducted was on the number of information-sharing communication links that were performed during the simulations. This analysis provides context for understanding the effectiveness of information sharing across the constellation, by showing the availability of information-sharing communications links as a function of constellation configuration.

Figure 11 shows an example of the communications links available and executed, in this case for the Walker constellation in the "Xlnk + Dlnk" case. Downlinks are always to a ground station, whereas crosslinks can be between two or more satellites. We focus here on four tightly crosslink-coupled satellites: 1, 4, 7, and 10. We observe that many of the windows are not executed, and there is a wide distribution of those that are. Notice that, for a lot of the executed crosslinks, not all of the possible crosslink partners actually perform the crosslink. This is particularly evident between satellites 7 and 10; they first succeed at crosslinking with each other at about 7 h. There is even one instance at about 4 h when satellite 7 tries to perform a crosslink with 4 and 10, but neither of the others joins in.

Table 5 summarizes the executed crosslinks, commlinks, and downlinks for the three constellations across all cases. We see that there are many more crosslink windows available for stitched Walker, at 89.2 versus 70.0 for plain Walker and 47.7 for ad hoc. We see that stitched Walker has much higher capability for sharing planning information via crosslinks. This conclusion makes sense, considering that "stitched" Walker was designed to maximize crosslink connectivity across the constellation. In the case with commlinks, the same number of commlinks (72) were available to each constellation (because they were spaced similarly for all constellations, every 20 min), and ad hoc executed a few more than the others on average (39.9 versus 34.4 and 33.3).

These results reveal that the constellations have a significant capacity for planning information sharing via crosslinks, but they do not necessarily use that access effectively. Recall that the RASP algorithm simply tries to execute all crosslinks and removes them in an essentially arbitrary fashion when the activity sequence cannot be scheduled. The algorithm currently does not consider whether the other possible partner satellites in the crosslink have decided to execute it nor how useful the crosslink could be for obtaining relevant planning information. Also, downlinks are only chosen by RASP to reduce data storage onboard the satellite [as indicated in Eq. (4)], not with consideration for planning information utility. In the current implementation, the crosslinks and downlinks chosen are those that happen not to interfere with observations, and not deliberately chosen based on their full utility.

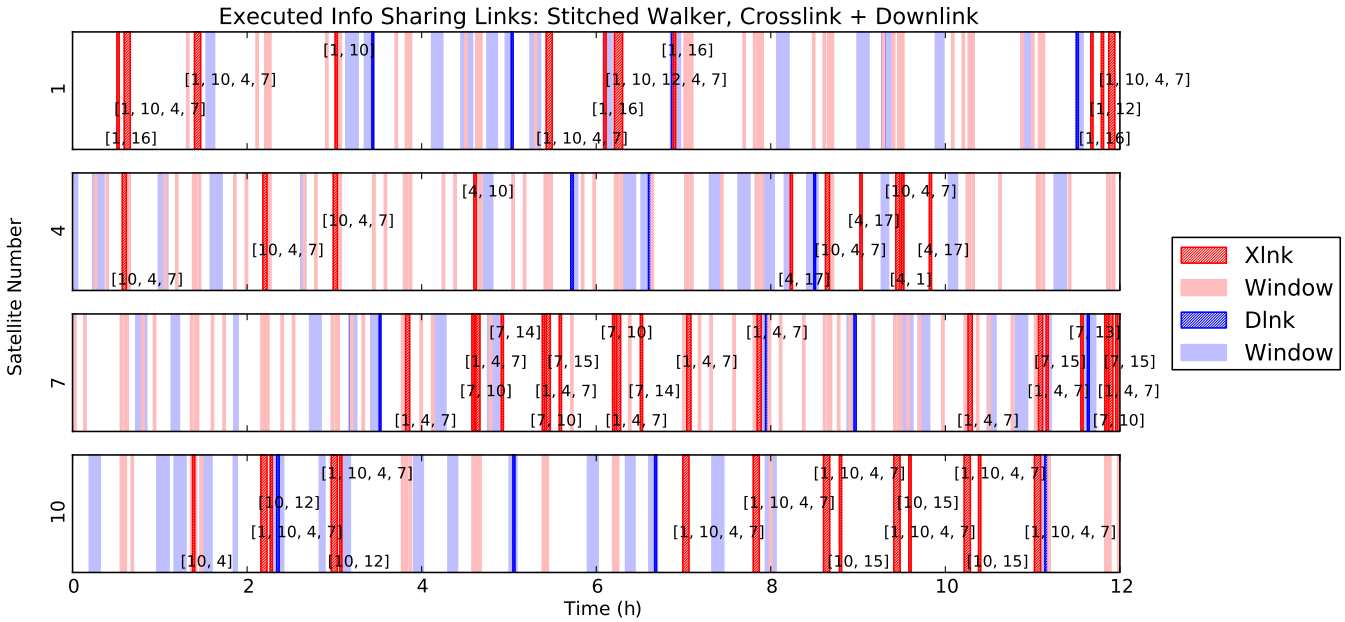

Fig. 11 Executed information sharing links in Xlnk + Dlnk case for stitched Walker constellation from beginning to 12 h. Only satellites 1, 4, 7, and 10 are included, for clarity. The labels in brackets correspond to the potential partner satellites for a crosslink; labels start at the bottom of a given satellite's plot, move up three slots, then wrap to bottom.

Table 5 Numbers of windows and executed crosslinks and commlinks, averaged over all satellites

Geometry	Item	Case		
		Xlnk	Xlnk + Dlnk	Clnk + Dlnk
Plain Walker	Number of windows	70.0	70.0	72.0
	Number executed	26.9	26.3	34.4
Stitched Walker	Number of windows	89.2	89.2	72.0
	Number executed	30.3	29.8	33.3
Ad hoc	Number of windows	47.7	47.7	72.0
	Number executed	20.8	20.7	39.9

C. Information Sharing Performance

The third analysis was performed on the effectiveness of planning information sharing across the constellation. Information sharing performance should depend on both communications link performance and orbital geometry: the more information sharing communications links performed, the faster and farther information will spread across the constellation.

Figures 12, 13 summarize the results for information sharing performance for the three constellations, in terms of the latency (CL) and timeliness (CIT) metrics introduced in Sec. II.E. The values in these plots are averaged over all "to–from" satellite combinations ($18 * 17$ values). Note that, for each combination, CL is averaged over all observations received by the receiving satellite, whereas CIT is averaged only over matching observations. As a reminder, matching observations must be of the same region and have starting times within a cutoff of 180 min (double the orbital period). Thus, the average CL values in the plots represent how long it takes information to move across the constellation as a whole, and the average CIT values represent how timely the received planning information was as a whole.

We see that CL and CIT are relatively low and high, respectively, in the Clnk + Dlnk case. This is as expected; the shared commercial communications backbone and ground network database gives the satellites easy access to each others' plans. Also, the values are good from a qualitative standpoint. Average CL is less than an orbit's duration (about 90 min). Average CIT is close to the length of the planning horizon (90 min), meaning that the satellites know about relevant planning information that could affect their observation choices roughly about the time

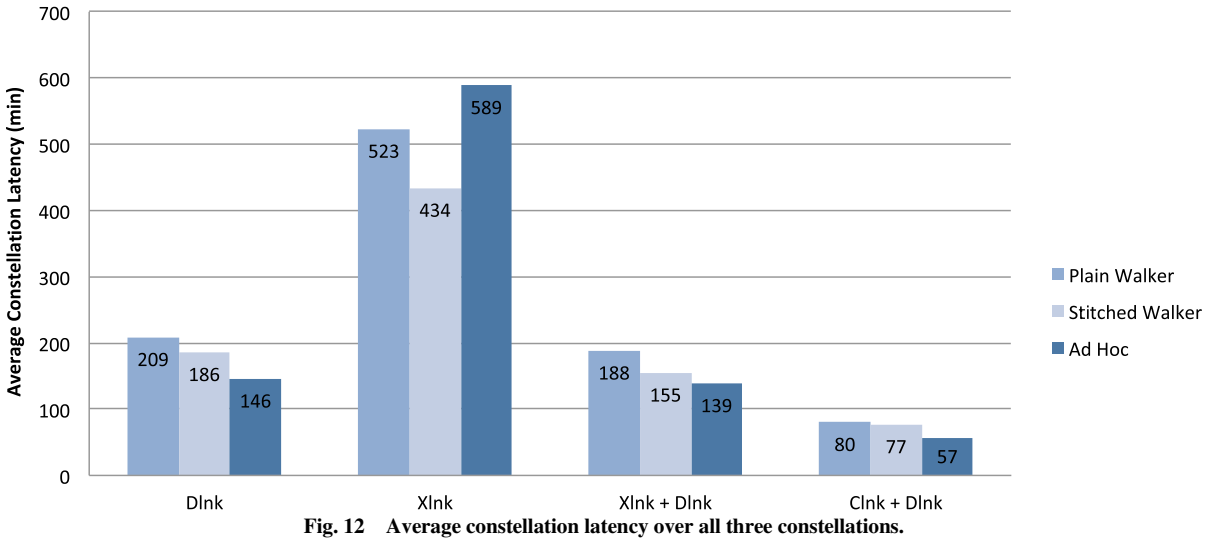

Fig. 12 Average constellation latency over all three constellations.

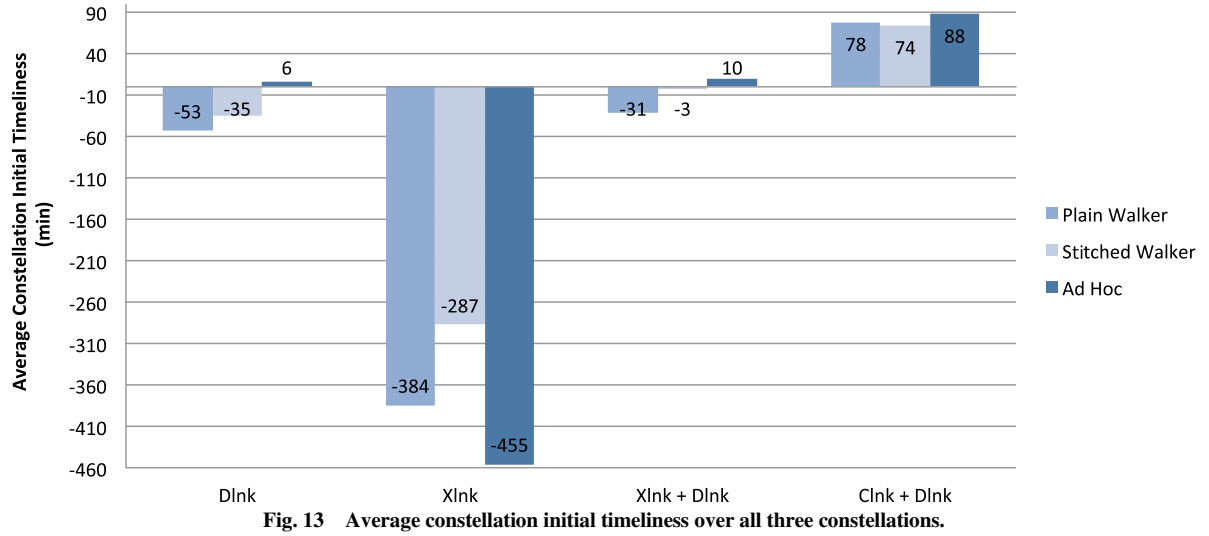

Fig. 13 Average constellation initial timeliness over all three constellations.

that they are first able to make those choices. But we see that performance is much worse in the other cases, particularly in Xlnk. For both the Dlnk and Xlnk + Dlnk cases, CL is on the order of two planning periods, and CIT hovers around 0 (the satellites get relevant planning information right around when they actually execute the observations affected). The Xlnk case has large negative CIT, indicating that the relevant planning information arrived much too late to be useful.

We can conclude from these results that, with the current versions of the RASP and LCCC algorithms, commlinks are currently the most effective means for information sharing, and downlinks are mildly effective. We also see that currently crosslinks are on average not very effective for information sharing across the constellation. Figures 14, 15 provide some more context for these conclusions. They show a breakout of CIT across all "from–to" satellite pairs for stitched Walker. In the Clnk + Dlnk case, we see that the vast majority of satellites receive timely planning information, with some small pockets of satellites that never receive any matching observations (indicated by red hashing pattern). This lack of matching observations for some pairs is caused by the fact that certain satellites never receive information about an observation of the same region within the same 180 min cutoff. For example, satellite 1 observes some of the same regions as 10, but only after 10 does. Thus, information exchanged between 10 and 1 tends to be more useful for 1. Ten never receives any relevant information from 1's plans. This fact is reflected in Fig. 14, which shows that the average CIT from 10 to 1 is 145 min, whereas from 1 to 10, there is nothing. In the Xlnk + Dlnk case (plot 15), the average CIT is much lower between most satellite pairs, reflecting the CIT value of −3 in Fig. 13 for stitched Walker.

Crosslink information sharing was likely ineffective for two reasons: many crosslinks were not executed by all partners, and the satellites by themselves do not have access to the shared planning information database. It appears to be much more difficult to share information effectively with crosslinks without specific deliberation about the utility of crosslinks and with so few satellites in the constellation. Interestingly, despite the

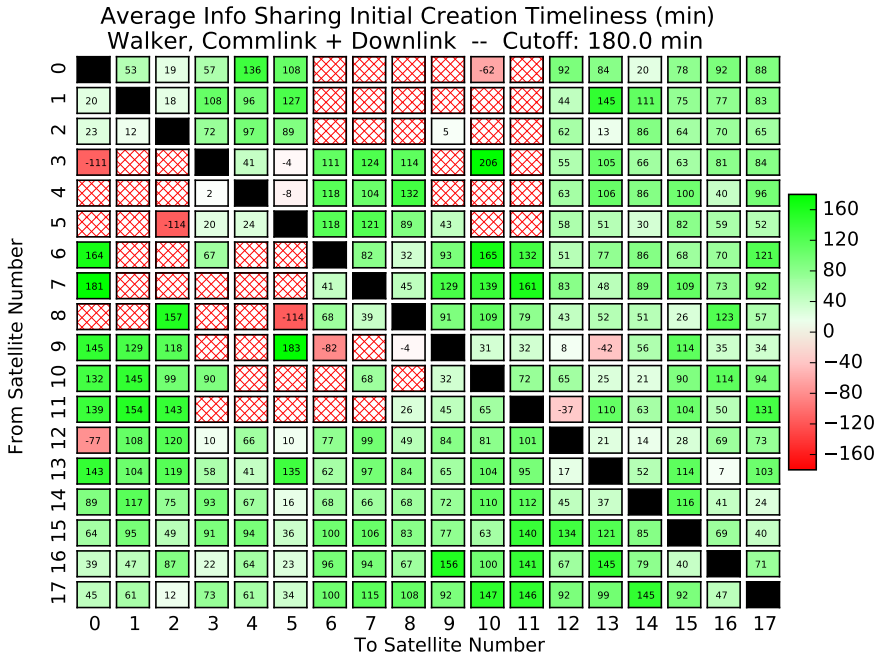

Fig. 14 Average constellation initial timeliness ("CIT") for stitched Walker star constellation, commlink + downlink case.

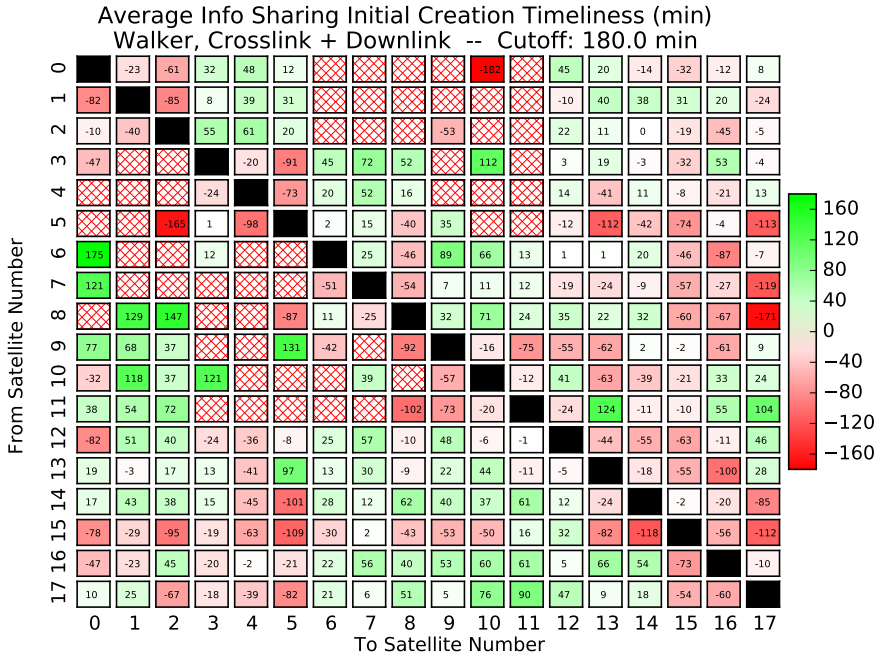

Fig. 15 Average constellation initial timeliness ("CIT") for stitched Walker star constellation, crosslink + downlink case.

low information sharing performance in the Xlnk case, Xlnk significantly outperformed Dlnk in terms of balancing average revisit times across sensors for stitched Walker (see Plot 9). Yet Dlnk outperformed Xlnk for plain Walker and ad hoc. It is unclear from the results how exactly this arises, but it may be because the crosslinks in stitched Walker happen to be more useful for observation planning than in the other constellations; remember that the stitched Walker geometry was designed specifically to aid in conveying information effectively across the constellation. More investigation of this behavior is merited in future work, perhaps with a better metric for assessing the planning utility of communications links.

16
17

VI. Conclusions

The performance in simulation of two algorithms for managing a coordinated CubeSat constellation with crosslinks is presented and analyzed: RASP and LCCC. The algorithms perform online, onboard planning for each satellite and use shared planning information over communications links to balance revisit times. The algorithms were designed to specifically address the needs of a widespread, coordinated constellation with infrequent communications links between neighbors. The present goals were 1) to demonstrate the application of these algorithms to distributed, online (real-time) prioritization of observations, 2) to assess their performance at balancing average revisit times across multiple observation regions and sensor types using various communications and planning strategies, and 3) to analyze the effectiveness of sharing planning information across the constellation with the different communications strategies.

The algorithms' performance was analyzed in a scenario with a set of 17 observation regions and three onboard sensor types. Three constellation geometries were investigated, plain Walker, stitched Walker and ad hoc, with several communications strategies: no information sharing, information sharing only through downlink, only through crosslink, through both crosslink and downlink, and through communication with a background constellation ("commlink") plus downlink. Performance was compared to a centralized planning strategy that selects sensors for the constellation's observations in advance using a simple greedy algorithm ("advance weighting") and another strategy that randomly selects sensors ("random selection"). The random selection strategy served as a base, no-coordination case against which to compare the algorithm's performance, and advance weighting served as a nearly ideal performance case that should be an upper limit for performance of online coordination (because it benefits from full global knowledge of satellites' possible observation activities).

RASP and LCCC were successfully applied in this scenario to increase observation revisit time performance. Comparing the commlink and downlink case to the random selection strategy, revisit time performance increased, with average revisit time values across all three sensors of 197, 203, and 225 min versus 204, 211, and 240 min (for plain Walker, stitched Walker and ad hoc, respectively; lower is better) and an intersensor range of within 11 min versus 28 min (between lowest and highest sensor revisit time; lower is better). Performance was best for the advance weighting strategy, with sensor-averaged revisit times of 194, 198, and 220 min and intersensor range of within 10 min. In the downlink and crosslink case, performance is mostly better than random selection, but not completely: the sensor-averaged revisit times of 199, 206, and 229 min are lower, but the intersensor range increased to 39 min. These results show that the observation coordination provided by the algorithms has a positive effect on revisit time performance as compared with simply randomly selecting observation sensors. But it is also concluded that the effect on performance is small; this specific revisit time use case does not benefit significantly from coordination. This is particularly true for the plain Walker constellation, where performance is very close between advance weighting, random selection, and the different communications and planning strategies (Fig. 8). These findings are relevant because they show that constellation coordination is indeed helpful, but it provides only a small performance boost for the specific observation operations concept in this work.

It was found that the stitched Walker constellation had many more crosslink opportunities per satellite than were available for plain Walker and ad hoc, at 89.2 versus 70.0 and 47.7 min, respectively. The stitched Walker constellation also executed more crosslinks per satellite in the case with only crosslink (30.3 versus 26.9 and 20.8) and appeared to benefit more from crosslinks, given that the crosslink case performed better than downlink for stitched Walker and not for the others. It was found that, with commlink and downlink, planned information is shared in a timely manner, with a CL (latency) of between 57 and 80 min and CIT (initial timeliness) of between 74 and 88 min across all constellation geometries. The CL and CIT dropped off precipitously in other cases and performed particularly poorly for only crosslink, with CL between 434 and 589 and CIT between -455 and -287. The poor information sharing performance in the downlink, crosslink, and crosslink plus downlink cases show that, in general, without external commlinks, information sharing performance across the constellation is poor. From Figs. 12, 13, it is seen that even the stitched Walker constellation, which was designed for good crosslink interconnectivity, performs poorly. Yet it is also concluded that revisit time performance is relatively insensitive to information sharing, based on the fact that this performance is good even in communications cases where information sharing is poor.

These results are a first step in assessing the utility of automated coordination via information sharing communications links for improving observation quality. It is seen that, for a simple objective like average revisit time performance, RASP and LCCC do improve results, but only by a small amount. This performance does not heavily depend of the efficacy of planning information sharing, though it does improve, as exemplified by the fact that performance is best in the commlink and downlink case. The results suggest that coordination would be more useful with an objective that requires tighter temporal and spatial coordination between the satellites (e.g., "sensor B must revisit sites visited by sensor A within 30 min for best performance"). They also suggest that some mechanism is needed in the algorithms for assessing the information sharing utility of communications links; the links should be weighted based on how helpful they are expected to be for finding out other satellites' plans. Although the results do not explicitly show a large benefit from coordination, they offer important guidance for further work in developing operations and automation strategies for coordinated constellations. Note also that, though the use of commlinks to an external backbone constellation may be too expensive and infeasible in practice, they provide a useful performance assessment point in this work.

These results should be repeatable because there is only a small stochastic element in the simulation, namely the small amount of noise on resource usage rates. The RASP and LCCC algorithms are aimed at CubeSats, and it is expected for them to be of relatively little aid in coordinating activities for less resource-constrained and communications-starved satellite constellations. Yet in their current form, they should be applicable to any size and orbit geometry of Earth-observing small satellite constellation, as long as the orbits are known well enough in advance (about 24 h) and command uplinks are available roughly on a daily basis.

There are several key limitations in the algorithms and performance analysis in this work. As discussed, the average revisit time minimization use case appears to not strongly depend on the effectiveness of information sharing. This metric was used as a simple first step for analyzing the potential of the RASP and LCCC algorithms and does indeed show that coordination improves performance versus random selection. The algorithms are clearly applicable to the current use case but would likely be more beneficial in a use case with more stringent observation coordination requirements.

The algorithms are naive in their handling of crosslinks, downlinks, and commlinks for information sharing; they simply perform as many as they can or are required to. As discussed, this limitation had only a mild effect on the revisit time performance. With more stringent observation coordination requirements, significant performance gains might be achieved by explicitly reasoning about and weighting the importance of these links for information sharing. Also, the commlinks are modeled very simply, with a window every 20 min. A more effective algorithm should be

able to exploit the continuous availability of a background constellation. The downlinks are modeled quite simply, too, not explicitly reasoning about downlink conflicts between satellites and using a percent overhead for the downlink rate.

The current implementation of RASP is only capable of scheduling a single onboard activity at a time, which sufficed in this work because only a single observation could occur at a time. Significant development, or the use of another planning algorithm, would be needed to generalize the planning to a simultaneous, multi-activity model while not overly increasing computational complexity. Observation windows are calculated with no regard for field-of-view restrictions, limiting the applicability of the algorithms presented here for general Earth-observation applications. The algorithms also do not reason about the importance or latency of the science data collected and how those data could be routed through the constellation.

There are several key refinements that are needed to improve the applicability and performance of the RASP and LCCC algorithms. LCCC should be refined to allow for more stringent temporal and spatial cooperation requirements between different satellites and observation targets. This might allow for optimizing other metrics of importance in Earth observation, such as age of information [58] of observations of targets or maximum windows of time and viewing angle overlap for observations. LCCC should also include a mechanism for assessing the information sharing utility of specific communications links, to maximize coordination performance. It is expected that further development of the algorithm in these two key areas will go a long way to making LCCC applicable to general Earth-observing SmallSat constellations. Other important updates to both RASP and LCCC include adding the ability to schedule multiple activities at once, adding observation field-of-view restrictions, managing data latency and routing through communications links, and modeling onboard energy consumption and production separately.

It is planned to extend this work to a more generally applicable layered planning system incorporating both ground (higher layer) and onboard planning (lower layer). This planning system will incorporate Earth-observation models with spatial and timing overlap requirements on observations as well as requirements on quantities and refresh rates of data delivered to ground. Also, more constellation geometries, perhaps with many more satellites, merit investigation for the increased network connectivity they could provide. It is hoped to integrate some degree of communications cost analysis, to assess the price of sharing information through crosslinks versus commlinks. A further investigation will be to examine the sensitivity of coordination results to the resource constraints on the CubeSats.

Another important item includes adapting the RASP code for implementation on an embedded spacecraft processor. The steps in this process would include translating the code to a lower-level language (e.g., C) and timing its performance on the processor. This work would help to verify the processing time and energy constraints due to the RASP planning process.

Appendix A:

Table A1 Parameters for plain Walker constellation

Satellite number	Altitude, km	Inclination, deg	RAAN, deg	True anomaly, deg	Eccentricity	Argument of perigee, deg
0	600	90	0	0	0	0
1	600	90	0	120	0	0
2	600	90	0	240	0	0
3	600	90	30	10	0	0
4	600	90	30	130	0	0
5	600	90	30	250	0	0
6	600	90	60	20	0	0
7	600	90	60	140	0	0
8	600	90	60	260	0	0
9	600	90	90	30	0	0
10	600	90	90	150	0	0
11	600	90	90	270	0	0
12	600	90	120	40	0	0
13	600	90	120	160	0	0
14	600	90	120	280	0	0
15	600	90	150	50	0	0
16	600	90	150	170	0	0
17	600	90	150	290	0	0

Table A2 Parameters for stitched Walker constellation

Satellite number	Altitude, km	Inclination, deg	RAAN, deg	True anomaly, deg	Eccentricity	Argument of perigee, deg
0	600	90	0	0	0	0
1	600	90	0	120	0	0
2	600	90	0	240	0	0
3	600	90	45	10	0	0
4	600	90	45	130	0	0
5	600	90	45	250	0	0
6	600	90	90	20	0	0
7	600	90	90	140	0	0
8	600	90	90	260	0	0
9	600	90	135	30	0	0
10	600	90	135	150	0	0
11	600	90	135	270	0	0
12	500	56	270	96	0	0
13	500	56	270	216	0	0
14	500	56	270	336	0	0
15	500	56	225	50	0	0
16	500	56	225	170	0	0
17	500	56	225	290	0	0

Table A3 Parameters for ad hoc constellation

Satellite number	Semimajor axis, km	Inclination, deg	RAAN, deg	True anomaly, deg	Eccentricity	Argument of perigee, deg
0	7128.14	51	0	110	0	0
1	7128.14	51	0	230	0	0
2	7128.14	51	0	350	0	0
3	7153.14	98	280	10	0	0
4	7153.14	98	280	130	0	0
5	7153.14	98	280	250	0	0
6	7078.14	98	235	35	0.01412801	0
7	7078.14	98	235	155	0.01412801	0
8	7078.14	98	235	275	0.01412801	0
9	7203.14	98	300	90	0	0
10	7203.14	98	300	210	0	0
11	7203.14	98	300	330	0	0
12	6978.14	52	10	30	0	0
13	6978.14	52	10	150	0	0
14	6978.14	52	10	270	0	0
15	7028.14	98	290	50	0	0
16	7028.14	98	290	170	0	0
17	7028.14	98	290	290	0	0

Table A4 Ground stations used in constellation simulations

Station	Number	Latitude, deg	Longitude, deg
Brazil (Goiania)	1	−16.71	−49.29
Fairbanks	2	64.84	−147.73
Germany (Stuttgart)	3	48.74	9.1
Hawaii	4	21.3	−157.86
Japan (Hakodate)	5	41.78	140.76
New England (Massachusetts Institute of Technology)	6	42.36	−71.09
New Zealand (Christchurch)	7	−43.54	172.73
Singapore	8	1.27	103.84
South Africa (Johannesburg)	9	−26.2	28.04

Table A5 Parameters for observation regions

Region number	Name	Latitude start, deg	Latitude end, deg	Longitude start, deg	Longitude end, deg
1	South America	−25	10	−80	−40
2	North Africa	4	36	−17	37
3	Southeast Asia	−10	19	92	150
4	Australia	−36	−14	113	152
5	East Asia	30	50	110	150
6	Europe	40	60	0	40
7	North America	25	55	−90	−60
8	Hawaii	15	35	−170	−150
9	Greenland	60	80	−60	−20
10	Russia	40	60	50	90
11	India	10	30	70	90
12	Western United States	30	50	−130	−110
13	South Africa	−40	−20	10	40
14	New Zealand	−30	−50	160	180
15	Alaska	50	70	−170	−140
16	Mideast	10	35	40	65
17	Argentina	−60	−30	−80	−60

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under grant number 1122374. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. This work is also supported by NASA Earth Science Technology Office grant numbers NNX14AC75G and NNX14AL95G and NASA Space Technology Research grant NNX12AM30H.

References

[1] "CubeSat Design Specification Rev. 13," California Polytechnic State Univ., TR, San Luis Obispo, CA, 2014.

[2] Rankin, D., Kekez, D. D., Zee, R. E., Pranajaya, F. M., Foisy, D. G., and Beattie, A. M., "The CanX-2 Nanosatellite: Expanding the Science Abilities of Nanosatellites," *Acta Astronautica*, Vol. 57, Nos. 2–8, 2005, pp. 167–174. doi:10.1016/j.actaastro.2005.03.032

[3] Springmann, J. C., Kempke, B. P., Cutler, J. W., and Bahcivan, H., "Development and Initial Operations of the RAX-2 CubeSat," *Proceedings of the 2012ESA/CNES Small Satellites Systems and Services Symposium*, No. 1, 2012, pp. 1–4.

[4] "Small Spacecraft Technology State of the Art," NASA TP-2014-216648, 2014.

[5] Klofas, B., "CubeSat Communication Systems: 2003-2015," http://www.klofas.com/comm-table/table.pdf [retrieved Feb. 2016].
[6] Wolfe, W. J., and Sorensen, S. E., "Three Scheduling Algorithms Applied to the Earth Observing Systems Domain," *Management Science*, Vol. 46, No. 1, 2000, pp. 148–166.
doi:10.1287/mnsc.46.1.148.15134
[7] Lematre, M., Verfaillie, G., Jouhaud, F., Lachiver, J. M., and Bataille, N., "Selecting and Scheduling Observations of Agile Satellites," *Aerospace Science and Technology*, Vol. 6, No. 5, 2002, pp. 367–381.
doi:10.1016/S1270-9638(02)01173-2
[8] Bensana, E., and Verfaillie, G., "Earth Observation Satellite Management," *Constraints*, Vol. 4, No. 3, 1999, pp. 293–299.
doi:10.1023/A:1026488509554
[9] Smith, B., Sherwood, R., Govindjee, A., Yan, D., Rabideau, G. R., Chien, S., and Fukunaga, A., "Representing Spacecraft Mission Planning Knowledge in ASPEN," TR, 1998.
[10] Rabideau, G., Knight, R., Chien, S., Fukunaga, A., and Govindjee, A., "Iterative Repair Planning for Spacecraft Operations Using the ASPEN System," *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Vol. 440, 1999, p. 99.
[11] Chien, S., Knight, R., Stechert, A., Sherwood, R., and Rabideau, G., "Using Iterative Repair to Increase the Responsiveness of Planning and Scheduling," *Proceedings of the 5th International Conf. on Artificial Intelligence Planning and Scheduling*, 2000, pp. 300–307.
[12] Chien, S., , Rabideau, G., Knight, R., Sherwood, R., Engelhardt, B., Mutz, D., Estlin, T., Smith, B., Fisher, F., and Barrett, T.et al., "ASPEN—Automated Planning and Scheduling for Space Mission Operations," *Proceedings of the International Conf. on Space Operations (SpaceOps 2000)*, Toulouse, France, 2000, pp. 1–10.
[13] Chien, S., Doubleday, J., Ortega, K., Tran, D., Bellardo, J., Williams, A., Piug-Suari, J., Crum, G., and Flatley, T., "Onboard Autonomy and Ground Operations Automation for the Intelligent Payload Experiment (IPEX) CubeSat Mission," *Proceedings of the International Symposium on Artificial Intelligence, Robotics, and Automation for Space*, Turin, Italy, 2012.
[14] Spangelo, S., and Cutler, J., "Optimization of Single-Satellite Operational Schedules Towards Enhanced Communication Capacity," *AIAA Guidance, Navigation, and Control Conf.*, AIAA Paper 2012-6410, Aug. 2012.
[15] Monmousseau, P., "Scheduling of a Constellation of Satellites: Improving a Simulated Annealing Model by Creating a Mixed-Integer Linear Model," M.S. Thesis, Royal Inst. of Technology, Stockholm, 2015.
[16] Surka, D. M., Brito, M. C., and Harvey, C. G., "The Real-Rime ObjectAgent Software Architecture for Distributed Satellite Systems," *Proceedings of the IEEE Aerospace Conf.*, Vol. 6, IEEE Publ., Piscataway, NJ, 2001, pp. 2731–2741.
[17] Schetter, T., Campbell, M., and Surka, D., "Multiple Agent-Based Autonomy for Satellite Constellations," *Artificial Intelligence*, Vol. 145, Nos. 1–2, 2003, pp. 147–180.
doi:10.1016/S0004-3702(02)00382-X
[18] Chien, S., , Sherwood, R., Burl, M., Knight, R., Rabideau, G., Engelhardt, B., Davies, A., Zetocha, P., Wainwright, R., and Klupar, P.et al., "The Techsat-21 Autonomous Sciencecraft Constellation," *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS 2001*, Saint-Hubert, QC, Canada, 2001.
[19] Chien, S., Engelhardt, B., Knight, R., Rabideau, G., Sherwood, R., Hansen, E. A., Ortiviz, A., Wilklow, C., and Wichman, S., "Onboard Autonomy on the Three Corner Sat Mission," *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation for Space*, 2001.
[20] Das, S., Wu, C., and Truszkowski, W., "Enhanced Satellite Constellation Operations via Distributed Planning and Scheduling," *Proceeding of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS 2001*, Saint-Hubert, QC, Canada, 2001.
[21] Van Der Horst, J., "Market-Based Task Allocation in Distributed Satellite Systems," Ph.D. Dissertation, Univ. of Southampton, Southampton, England, U.K., 2012.
[22] Van der Horst, J., and Noble, J., "Task Allocation in Networks of Satellites with Keplerian Dynamics," *Acta Future*, Vol. 5, 2012, pp. 143–151.
[23] Damiani, S., Verfaillie, G., and Charmeau, M.-C., "An Earth Watching Satellite Constellation: How to Manage a Team of Watching Agents with Limited Communications," *Proceedings of the 4th International Joint Conf. on Autonomous agents and Multiagent Systems — AAMAS '05*, Utrecht, Netherlands, 2005, p. 455.
[24] Wang, D., and Williams, B. C., "tBurton: A Divide and Conquer Temporal Planner," TR, 2014, http://dspace.mit.edu/bitstream/handle/1721.1/91170/MIT-CSAIL-TR-2014-027.pdf?sequence=2 [retrieved _____].
[25] Wang, D., and Williams, B., "tBurton: Model-Based Temporal Generative Planning," *KISS Workshop: Engineering Resilient Space Systems: Leveraging Novel System Engineering Techniques and Software Architectures*, Pasadena, CA, 2012.
[26] Amato, C., Chowdhary, G., Geramifard, A., Ure, N. K., and Kochenderfer, M. J., "Decentralized Control of Partially Observable Markov Decision Processes," *Proceedings of the 52nd IEEE Conf. on Decision and Control*, IEEE Publ., Piscataway, NJ, 2013, pp. 2398–2405.
[27] Amato, C., Konidaris, G. D., and Kaelbling, L. P., "Planning with Macro-Actions in Decentralized POMDPs," *Proceedings of the Workshop on Planning and Robotics (PlanRob) at the 24th International Conf. on Automated Planning and Scheduling (ICAPS-14)*, Portsmouth, NH, 2014, pp. 1273–1280.
[28] Choi, H. L., Brunet, L., and How, J. P., "Consensus-Based Decentralized Auctions for Robust Task Allocation," *IEEE Transactions on Robotics*, Vol. 25, No. 4, 2009, pp. 912–926.
doi:10.1109/TRO.2009.2022423
[29] Kennedy, A., , Marinan, A., Cahoy, K., Byrne, J., Cordeiro, T., Decker, Z., Marlow, W., Blackwell, W. J., Diliberto, M., and Leslie, R. V.et al., "Automated Resource-Constrained Science Planning for the MiRaTA Mission," *Proceedings of the 29th Annual AIAA/USU Conf. on Small Satellites*, Logan, UT, 2015, pp. SSC15–VI–1.
[30] Kennedy, A. K., and Cahoy, K. L., "Onboard Operations Scheduling for a Cooperative Earth Remote Sensing Small Satellite Constellation," *Proceedings of the 9th Annual International Workshop on Spacecraft Constellations and Formation Flying*, Delft, The Netherlands, 2015.
[31] Kennedy, A. K., "Resource Optimization Algorithms for an Automated Coordinated CubeSat Constellation by," M.S. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, 2015.
[32] Kennedy, A., and Cahoy, K., "The MiRaTA CubeSat FSW Architecture & Scaling CubeSat FSW to Cooperative Constellations," *Workshop on Spacecraft Flight Software*, Pasadena, CA, 2014.
[33] "Iridium Global Network," Iridium Communications [retrieved Feb. 2016].
[34] "Constellation," Globalstar [retrieved Feb. 2016].
[35] Castaing, J., "Scheduling Downloads for Multi-Satellite, Multi-Ground Station Missions," *Proceedings of the 28th Annual AIAA/USU Conf. on Small Satellites*, Logan, UT, 2014, pp. 1–12.
[36] Orfanidis, S. J., "Linear and Loop Antennas," *Electromagnetic Waves and Antennas*, edited by Orfanidis, S. J., 2001, p. 778, Chap. 17, http://www.ece.rutgers.edu/{~} orfanidi/ewa [retrieved _____].
[37] Coffee, B. G., Cahoy, K., and Bishop, R., "Propagation of CubeSats in LEO Using NORAD Two Line Element Sets: Accuracy and Update Frequency," *AIAA Guidance, Navigation, and Control Conf.*, AIAA Paper 2013-4944, 2013.
[38] Riesing, K., "Orbit Determination from Two Line Element Sets of ISS-Deployed CubeSats," *Proceedings of the 29th Annual AIAA/USU Conf. on Small Satellites*, 2015, p. VIII–5.
[39] Gangestad, J. W., Hardy, B. S., and Hinkley, D. A., "Operations, Orbit Determination, and Formation Control of the AeroCube-4 CubeSats," *Proceedings of the 27th Annual AIAA/USU Conf. on Small Satellites*, Logan, UT, 2013.
[40] Greene, M. R., and Zee, R. E., "Increasing the Accuracy of Orbital Position Information from NORAD SGP4 Using Intermittent GPS Readings," *Proceedings of the 23rd Annual AIAA/USU Conf. on Small Satellites*, Logan, UT, 2009.
[41] Bertsimas, D., and Weismantel, R., *Optimization over Integers*, Vol. 13, Dynamic Ideas, Belmont, MA, 2005.
[42] Walker, J., "Satellite Constellations," *Journal of the British Interplanetary Society*, Vol. 37, 1984, pp. 559–571.

34

[43] Walker, J., "Circular Orbit Patterns Providing Continuous Whole Earth Coverage," Royal Aircraft Establishment, TR, 1970.
[44] Legge, R. S., "Optimization and Valuation of Reconfigurable Satellite Constellations Under Uncertainty," Ph.D. Dissertation, Massachusetts Inst. of Technology, Cambridge, MA, 2014.
[45] Marinan, A., Nicholas, A., and Cahoy, K., "Ad Hoc CubeSat Constellations: Secondary Launch Coverage and Distribution," *Proceedings of the IEEE Aerospace Conf.*, IEEE Publ., Piscataway, NJ, 2013.
[46] Ballard, A., "Rosette Constellations of Earth Satellites," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 16, No. 5, 1980, pp. 656–673. doi:10.1109/TAES.1980.308932
[47] Hanson, J. M., and Lindenj, A. N., "Improved Low-Altitude Constellation Design Methods," *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 2, 1989, pp. 228–236. doi:10.2514/3.20395
[48] Razoumny, Y. N., "Analytic Solutions for Earth Discontinuous Coverage and Methods for Analysis and Synthesis of Satellite Orbits and Constellations," *AIAA/AAS Astrodynamics Specialist Conf.*, AIAA Paper 2014-4160, Aug. 2014.
[49] Nag, S., Lemoigne, J., Miller, D. W., and De Weck, O., "A Framework for Orbital Performance Evaluation in Distributed Space Missions for Earth Observation," *Proceedings of the IEEE Aerospace Conf.*, Vol. 2015, IEEE Publ., Piscataway, NJ, June 2015.
[50] Blackwell, W. J., , Allen, G., Galbraith, C., Leslie, R., Osaretin, I., Scarito, M., Shields, M., Thompson, E., Toher, D., and Townzen, D.et al., "MicroMAS: A First Step Towards a Nanosatellite Constellation for Global Storm Observation," *Proceedings of the 27th Annual AIAA/USU Conf. on Small Satellites*, Logan, UT, 2013, p. XI–1.
[51] Blackwell, W. J., , Allan, G., Allen, G., Burianek, D., Busse, F., Elliott, D., Galbraith, C., Leslie, R., Osaretin, I., and Shields, M.et al., "Microwave Radiometer Technology Acceleration Mission (MiRaTA): Advancing Weather Remote Sensing with Nanosatellites," *Proceedings of the 28th Annual AIAA/USU Conf. on Small Satellites*, Logan, UT, 2014, pp. P4–12.
[52] Cahoy, K. L., Marinan, A., Marlow, W., Cordeiro, T., Blackwell, W. J., Bishop, R., and Erickson, N., "Development of the Microwave Radiometer Technology Acceleration (MiRaTA) Cubesat for All-Weather Atmospheric Sounding," *Proceedings of IGARSS 2015*, Milan, 2015, pp. 5304–5307.
[53] "BeagleBone Black System Reference Manual," Beagleboard, http://www.adafruit.com/datasheets/BBB_SRM.pdf [retrieved Aug. 2015].
[54] "Raspberry Pi 2 Model B Webpage," Raspberry Pi Foundation, https://www.raspberrypi.org/products/raspberry-pi-2-model-b [retrieved Aug. 2015].
[55] Boshuizen, C. R., Mason, J., Klupar, P., and Spanhake, S., "Results from the Planet Labs Flock Constellation," *Proceedings of the 28th Annual AIAA/USU Conf. on Small Satellites*, Logan, UT, 2014, p. I-1.
[56] Fish, C., Swenson, C., Neilsen, T., Bingham, B., Gunther, J., Stromberg, E., Burr, S., Burt, R., Whitely, M., and Petersen, J., "DICE Mission Design, Development, and Implementation: Success and Challenges," *Proceedings of the 26th Annual AIAA/USU Conf. on Small Satellites*, Logan, UT, 2012.
[57] "linprog Documentation," MATLAB, http://www.mathworks.com/help/optim/ug/linprog.html [retrieved Feb. 2016].
[58] Kaul, S., Yates, R., and Gruteser, M., "Real-Time Status: How Often Should One Update?" *Proceedings of the 31st Annual IEEE International Conf. on Computer Communications: Mini-Conf.*, IEEE Publ., Piscataway, NJ, 2012, pp. 2731–2735.

A. Golkar
Associate Editor

35
36
37

Queries

1. AU: Please check that the copyright (©) information is correct.
2. AU: Please check that the authors' affiliations and footnotes are correct.
3. AU: Acronyms are not allowed in the Abstract.
4. AU: The first section must be called "Introduction".
5. AU: Acronyms that are defined and used only once (e.g., IPEX, LP) are removed.
6. AU: Please provide appropriate reference citation for "Gombolay, Wilcox, and Shah".
7. AU: AIAA has not indicated color use for your paper. Please confirm that your paper should be grayscale and that all figures are satisfactory. If any replacement figures are needed, please send them in .eps or .tiff format to aiaa-proofs@beacon.com and use code I010426 in the subject line. Formats of .jpg, .doc, or .pdf can be used with some loss of quality.
8. AU: All tables must be cited in numerical order in the main text. Various tables were cited out of order. Please review table citations, and modify them so that all tables are cited in numerical order.
9. AU: All references must be cited in numerical order in the main text. Reference [32] was cited in the caption to Fig. 2 but not in the main text. A citation of this reference was added to the main text near the citation of Fig. 2. Please confirm that your intended meaning is retained.
10. AU: All figures must be cited in numerical order in the main text. Many figures were cited out of order. Please review figure citations, and modify them so that all figures are cited in numerical order.
11. AU: Section headings cannot contain acronyms or begin with articles (e.g., "the").
12. AU: The captions to Figs. 4–7, 11 exceed the maximum length of 30 words. Please reduce them. You may consider moving some information to the main text.
13. AU: The unit of measure "AU" is not used by AIAA. Please spell out or define this unit of measure.
14. AU: Should "discluded" (as used in the captions to Figs. 6, 7) instead be "excluded"?
15. AU: Because figures will not appear in color, please rewrite or remove from the text all references to color in the figures (e.g., "red hashing pattern").
16. AU: The Conclusions section cannot have subsection headings. These headings have been removed.
17. AU: First person is not allowed in the Conclusions. Please confirm that your intended meaning is retained.
18. AU: Please provide a short title for the Appendix.
19. AU: The figures in the Appendix were changed to tables. Please confirm that your intended meaning is retained.

20. AU: Please spell out "RAAN" in Tables 1–3, or provide the definition in the text.

21. AU: Please provide the report number for Ref. [1].

22. AU: If Refs. [3, 10, 11, 23, 27, 35, 38–40, 52] are published proceedings, please provide the name and location of the publisher (NOT of the conference host).

23. AU: URLs in references must be accompanied by name and location of publisher as well as date of publication. Please provide this information for Ref. [5]. If this information is not available, please restyle as a footnote: "Data available online at ___URL___ [retrieved ___DATE___]."

24. AU: Please provide the name and location of the institution as well as the report number for Ref. [9].

25. AU: Please provide all author names for Refs. [12, 50, 51]. Furthermore, if they are published proceedings, please provide the name and location of the publisher (NOT of the conference host).

26. AU: If Refs. [13, 19, 20, 30, 32, 49, 55, 56] are published proceedings, please provide the name and location of the publisher (NOT of the conference host) and the page numbers used. If they are conference papers, please provide the paper number and the organizer's name. If they are CD-ROMs, please provide the name and location of the CD-ROM producer.

27. AU: Please provide all author names for Refs. [18, 29]. Furthermore, if they are published proceedings, please provide the name and location of the publisher (NOT of the conference host) and the page numbers used. If they are conference papers, please provide the paper number and the organizer's name.

28. AU: Please provide the issue number (if applicable) and/or the month of publication for Ref. [22].

29. AU: URLs in references must be accompanied by name and location of publisher as well as date of publication and date of retrieval. Please provide this information for Ref. [24]. If this information is not available, please restyle as a footnote: "Data available online at ___URL___ [retrieved ___DATE___]."

30. AU: Please provide additional information for Ref. [25]. For example, if it is a book, please provide name and location of publisher and pages used. If it is a published proceedings, please provide the name and location of the publisher (NOT of the conference host) and the page numbers used. If it is a conference paper, please provide the paper number and the organizer's name.

31. AU: Please provide location of publisher, date of publication, and valid URL for Refs. [33, 34].

32. AU: The URL provided for Ref. [36] does not work. Please provide a valid URL as well as name and location of publisher and date of retrieval for the URL.

33. AU: Please provide the pages used for the book in Ref. [41].

34. AU: Please provide the issue number (if applicable) and/or the month of publication for Ref. [42].

35. AU: Please provide the location for Ref. [43] as well as the report number.

36. AU: Please provide the pages for Ref. [45].

37. AU: Please provide the location of the publisher and date of publication for the URLs in Refs. [53, 54, 57].