On the Automation, Optimization, and In-Orbit Validation of Intelligent Satellite Constellation Operations

Gregory Stock, Juan A Fraire, Holger Hermanns, Eduardo Cruz, Alastair Isaacs, Zhana Imbrosh

[SSC21-V-05]

On the Automation, Optimization, and In-Orbit Validation of Intelligent Satellite Constellation Operations

Gregory Stock
Saarland University
Saarland Informatics Campus E1 3, 66123 Saarbrücken, Germany; +49 681 302 5635
stock@depend.uni-saarland.de

Juan A. Fraire
Saarland University / INRIA
Saarland Informatics Campus E1 3, Saarbrücken, Germany; +54 9351 244 6010
INSA Lyon, 20 Avenue Albert Einstein, 69621 Villeurbanne CEDEX, France
juanfraire@depend.uni-saarland.de

Holger Hermanns
Saarland University / Institute of Intelligent Software
Saarland Informatics Campus E1 3, 66123 Saarbrücken, Germany; +49 681 302 5631
Institute of Intelligent Software, Jinzhu Plaza, No. 221 West Huanshi Avenue, Nansha, Guangzhou, China
hermanns@depend.uni-saarland.de

Eduardo Cruz, Alastair Isaacs, Zhana Imbrosh
GomSpace
GomSpace A/S, Langagervej 6, 9220 Aalborg Øst, Denmark; +352 621 291 207
ecru@gomspace.com

ABSTRACT

Recent breakthroughs in technology have led to a thriving "new space" culture in low-Earth orbit (LEO) in which performance and cost considerations dominate over resilience and reliability as mission goals. These advances create a manifold of opportunities for new research and business models but come with a number of striking new challenges. In particular, the size and weight limitations of low-Earth orbit small satellites make their successful operation rest on a fine balance between solar power infeed and the power demands of the mission payload and supporting platform technologies, buffered by on-board battery storage. At the same time, these satellites are being rolled out as part of ever-larger constellations and mega-constellations. Altogether, this induces a number of challenging computational problems related to the recurring need to make decisions about which task each satellite is to effectuate next. Against this background, GOMSPACE and Saarland University have joined forces to develop highly sophisticated software-based *automated* solutions rooted in *optimal algorithmic* and *self-improving learning techniques*, all this *validated* in modern nanosatellite networked missions operating in orbit.

The paper introduces the GOMSPACE Hands-Off Operations Platform (HOOP), an automated, flexible, and scalable end-to-end satellite operation framework for commanding and monitoring subsystems, single-satellites, or constellation-class missions. To this, the POWVER initiative at Saarland University has contributed state-of-the-art dynamic programming and learning techniques based on profound battery and electric power budget models. These models are continually kept accurate by extrapolating data from telemetry received from satellites. The resulting machine learning approach delivers optimal, efficient, scalable, usable, and robust flight plans, which are provisioned to the satellites with zero need for human intervention—but which are still under the full control of the mission operator. We report on insights gained while validating the integrated POWVER-HOOP approach in orbit on the dual-satellite mission GOMX–4 by GOMSPACE that is currently in orbit.

INTRODUCTION

Near-Earth satellites are being launched by the thousands; an unprecedented pace made possible by recent breakthroughs in technology accompanying a "new space" culture where cost/performance considerations dominate over resilience/reliability (i. e., emergence of COTS components and CubeSat platforms). Although these advances create many opportunities for new research and business models, a number of striking new challenges need to be tackled in order to efficiently manage the available resources while also ensuring maximum payload utilization. In particular, the size and weight limitations of low-Earth orbit (LEO) small satellites mean that their successful operation rests on a fine balance between solar power infeed and the power demands of the mission payload and supporting platform technologies, buffered by on-board battery storage. This renders a non-evident, recurring, and intricate scheduling problem to be solved on the ground segment, namely the continual need to make decisions about which task the satellite is to effectuate next. This requirement will arguably become the bottleneck for the growing trend of scaling the space segment to constellations and mega-constellations.

To this end, we contribute sophisticated software-based *automated* solutions rooted in *optimal* computer science techniques *validated* in modern nano-satellite networked missions operating in orbit. This paper first introduces the GOMSPACE Hands-Off Operations Platform (HOOP), a flexible and scalable end-to-end satellite operation framework for commanding and monitoring subsystems, single-satellites, or constellation-class missions. By taking advantage of the vast expertise of GOMSPACE, new space actors can leverage flight-proven toolchains throughout the mission lifecycle while profiting from partner ground station networks without the need to invest in their own operational infrastructure. Second, we present how HOOP is enhanced by highly efficient and accurate automated decision-making capabilities exploiting dynamic programming and learning techniques based on profound battery and electric power budget models, developed at Saarland University as part of the POWVER initiative. The models are continually kept accurate by extrapolating data from telemetry received from satellites. The resulting machine learning approach delivers optimal, efficient, scalable, usable, and robust flight plans, which are provisioned to the satellites with zero need for human intervention—but which are still under the full control of the mission operator. Third, we report on the application of the POWVER-HOOP approach to GOMX–4, the dual-satellite mission by GOMSPACE that is currently in orbit. Over a period of more than a month, a series of in-orbit experiments have been carried out with the 6U CubeSats, covering Earth observation, air traffic surveillance, as well as inter-satellite linking capabilities. In these experiments, the integrated POWVER-HOOP toolchain has shown its unique strength, namely to operate a mission without human intervention while persistently delivering maximum return from its observation payloads and ensuring the most efficient and safe utilization of constrained on-board battery resources. We make these findings concrete by reporting details of a 48-hour period selected from the masses of recorded experimental results.

This pioneering work evidences that humans can define and supervise high-level objectives of the mission while relying on machine learning approaches to finally unblock the future of space operations.

CONTEXT

GOMSPACE and Their Mission

Since the foundation of the company in 2007, GOMSPACE has become a leading manufacturer and supplier of CubeSats and small satellite solutions for customers in academic, government, and commercial markets. The key strengths of the company include systems integration, CubeSat platforms, advanced miniaturized radio technology, and satellite operations. The GOMSPACE headquarters are located in Aalborg, Denmark. The company also has a propulsion technology center in Uppsala, Sweden, and a satellite operations center in Esch-sur-Alzette, Luxembourg. The company currently employs more than 150 people and provides services to customers in more than 60 nations.

GOMSPACE has a track record of successful missions in space. This is exemplified by the GOMX series of satellites, all of which were built and operated by GOMSPACE. GOMX–1, a 2U satellite launched in November 2013, successfully demonstrated for the first time the reception of ADS-B signals from aircraft by an orbiting satellite. The satellite remains in orbit. GOMX–3, launched in 2015, demonstrated attitude control, downlinking of data, and SATCOM spot-beam characterization. The satellite successfully completed its nominal mission and re-entered the atmosphere after one year. That mission was followed by the GOMX–4 mission, a pair of two 6U CubeSats. This mission demonstrated the ability of CubeSats to act in coordination through inter-

satellite communication. Payloads of the satellites are used for surveillance and monitoring of Arctic regions.

These projects are delivered based on GOMSPACE's strong in-house portfolio of established products and rich capabilities. Currently, GOMSPACE is developing the Juventas nanosatellite that will form part of ESA's Hera mission to the Didymos binary asteroid system.[1] The mission will provide valuable scientific data from the asteroid system, including radar and radio science observations of the binary system.

With more than 13 years of experience in the market and a track record of multiple successful missions accomplished, GOMSPACE has developed profound knowledge and competencies within radio technology, CubeSat platforms, project management and innovation. Starting early as a pioneer in the market, GOMSPACE has now become a market leader in the commercialization of nanosatellites and new space technology.

The numerous lessons learned by GOMSPACE during past satellite projects and operations materialize into HOOP. Described in detail below, HOOP is a platform enabling automated satellite operation at low cost that is individually adapted to the specific mission requirements. As evidence of HOOP's flexibility, this paper presents its seamless integration with the LEOPOWVER toolchain.

POWVER at Saarland University

Since 2013, Saarland University is performing scientific research on the operation of nanosatellites, seeded in the EU FP7 project SENSATION*, where GOMSPACE acted as an industrial partner. The unique expertise of the Saarland University researchers lies in the application of formal methods to perform automatic and resource-optimal task scheduling of satellites and satellite constellations. Early work developed the scientific grounds along the GOMX–1[2] and GOMX–3 missions, including in-orbit demonstrations[3,4]. In these contexts, it had become apparent that there is massive room for improvement by properly modeling and analyzing the satellite's battery, operational constraints, and orbital environment. The research intensified as a consequence of a multi-million dollar award, the ERC Advanced Grant POWVER†, that was awarded in 2016 to Holger Hermanns by the European Research Council to foster the research. Another (albeit smaller) grant, the ERC Proof of Concept Grant LEOPOWVER‡, is nowadays the focus point of all application and commercialization activities of what has been developed successfully over the years: *orbit-proof software enabling the continuous, fully automated, energy-optimal, and profit-maximizing dynamic operation of LEO satellites and satellite constellations.*

At the core of the LEOPOWVER software is a unique collection of highly realistic battery models that enable efficient forecasting of battery health and battery depletion risk with unprecedented accuracy. This is paired with ultra-efficient optimization and machine learning techniques that are tailored to the LEO context, including dedicated support for telemetry processing and contact plan design for satellite constellations.[5] Indeed, communication transmitters and transponders are among the most power-demanding subsystems of any modern spacecraft. This phenomenon is exacerbated in networked space constellations supported by one or more Inter-Satellite Link (ISL) interfaces. As a result, a notorious bottleneck is provoked by power-hungry networking tasks that need to be powered by constrained batteries and solar power infeed. This asks for a very careful time-evolving and data-driven scheduling of the communication resources.

The methodologies delivered by LEOPOWVER are targeted at the core of this problem, which needs to be perpetually solved during the mission lifetime. Thus, the role of the operator is reduced to the most important aspects of the mission: defining the goal and the conditions to achieve it, leaving computer science algorithms to ensure the optimal and secure control of the space system. LEOPOWVER harvests very advanced algorithmic and learning approaches, which are already unlocking the optimal battery-aware operations of future cross-linked satellite constellations.[6,7]

The spirit of LEOPOWVER is to make this unique combination of technologies ready for take-up by embracing start-ups, academics, and business customers with the intention to create a prospering user community. Thus, favorable conditions will be enabled via a base version of the software to be released under open-source licensing in the near future.§ To this end, the LEOPOWVER orchestration toolchain is designed to provide highly flexible support for integration into arbitrarily complex operations workflows via well-defined telemetry and commanding interfaces across the entire spectrum of orbit applications.

*`https://cordis.europa.eu/project/id/318490`
†`https://cordis.europa.eu/project/id/695614`
‡`https://cordis.europa.eu/project/id/966770`
§Check `https://leopowver.space/` for the latest updates.

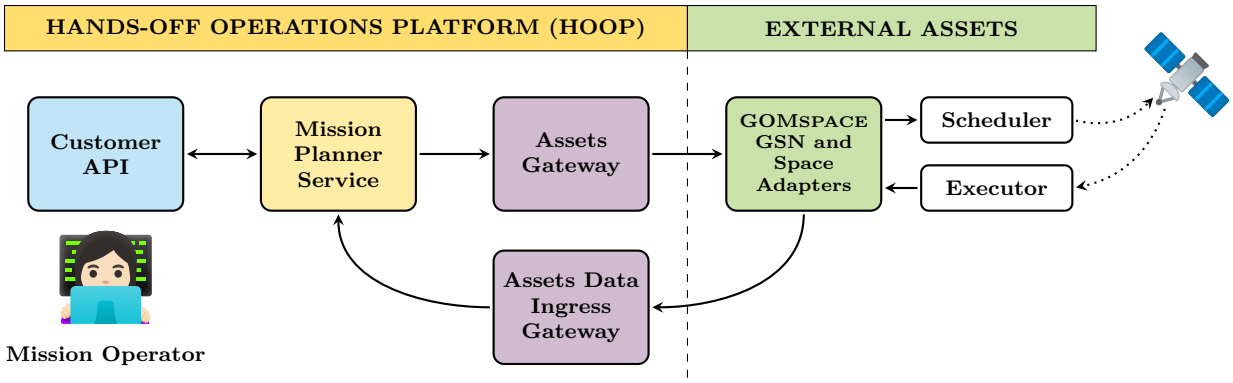

Figure 1: Overview of the GOMSPACE Hands-off Operations Platform (HOOP).

In this paper, we describe how this flexibility is exploited for a straightforward and successful integration of LEOPOWVER with the GOMSPACE HOOP operations platform to then validate its applicability in the state-of-the-art GOMX–4 nanosatellite mission currently in orbit.

HANDS-OFF OPERATIONS PLATFORM

The GOMSPACE Hands-off Operations Platform (HOOP) is a satellite operations platform built for automation, scalability, and flexibility. The HOOP platform was developed by GOMSPACE specifically to handle operations for constellations of CubeSats, up to constellations with thousands of satellites. Since 2018, the platform has been under development with support from the Luxembourg Space Agency (LSA) and European Space Agency (ESA).

Unlike traditional operations centers, which carry out operations manually with large numbers of satellite operators, operations centers for constellations must automate much of the nominal operations. The HOOP system has been designed to support and manage this degree of operations, and thereby allow operations to scale to support hundreds or thousands of satellites.

In this way, HOOP streamlines routine operations, allowing operators to focus on troubleshooting and improving the mission. This reduces the manual effort required to monitor and maintain each satellite, and also means that routine activities, such as payload data capture and download, can be completely automated and require no manual input at all.

HOOP provides a set of distinguishing features for the management and autonomous operation of satellites, from the first satellite to a global constellation. A schematic overview displaying all components that HOOP consists of is shown in Figure 1.

Configuration Management. A difficult challenge in satellite operations is to track and know the satellite configuration at all times, even when the satellite is out of visibility of a ground station. HOOP addresses this challenge by providing a set of configuration management tools. The last-known satellite configuration is stored in a database and visible to operators through a user interface. All changes planned through HOOP are recorded, and all downlinked telemetry is monitored for any discrepancies with the expected configuration. Operators are alerted if any unexpected configuration changes are detected. The database used by HOOP allows the configuration of multiple satellites to be tracked. When an issue arises with a satellite, an operator can call up the relevant satellite from the database and review the last known configuration of all on-board parameters.

Mission Planning. The operations software handles both manual and automated mission planning. Operators can define routine procedures that are scheduled according to specified rules (based on events or time). They can also plan manual procedures as required, expressed in a high-level yet flexible procedure language. A plan resolver checks all operations plans, generated either manually or automatically, and verifies that there are no conflicts present. If conflicts are found, the resolver attempts to resolve them. If no solution can be found, it will reject the operations plan and send an alert to the operator.

Contact Scheduling. After procedures are validated, they are scheduled for upload and execution on the satellite. HOOP can handle multiple ground stations and will schedule the upload on the next available pass for the satellite over any of the available ground stations. Once the pass begins, the scheduled commands are executed, and prepared files are uploaded. The command status is always visible to the operator, so they are aware if the commands are planned, have been executed, or failed. If a command fails to execute on the satellite or a file upload fails, HOOP can be configured to either abandon the effort or retry.

Payload Data Download. The download of collected payload data from the satellite is scheduled autonomously by HOOP. The best opportunities for downloading the data are calculated, considering any conflicts and resource availability (i. e., available data storage). After new data is downloaded, HOOP transfers it to the end user.

Telemetry Handling. Telemetry received from the satellite is stored in a telemetry database. Operators can visualize real-time and historical telemetry data using HOOP's telemetry dashboard. This dashboard offers a flexible and easily configurable view of the data. Operators can select the data they want to see, and can quickly request data from different time periods and with different granularity. This telemetry data is also visible to users through the customer API (application programming interface).

Alarms. Operators can define rules for raising alarms. HOOP constantly monitors the incoming telemetry data and will raise alarms when the telemetry meets conditions defined in those rules. Operators are informed through the platform and through notifications, which can be sent via email or to mobile devices. An alarm console records all active alarms. Operators can also define rules for automatically responding to specific alarms. This feature will only be used for alarms that are well understood and have a known resolution. All automatically taken commands are logged and available for review by an operator.

Flight Dynamics. HOOP uses Orekit[8] for internal flight dynamics calculations, including the calculation of ground station passes and orbital events. HOOP is also compatible with tools developed in Orekit or GMAT[9] for maneuver planning, e. g., for formation deployment, station keeping, or collision avoidance.

Simulation. HOOP incorporates a number of simulation tools for mission planning. This includes data, thermal, and link simulators. These tools analyze the current satellite state, based on the telemetry and known configuration, and generate forecasts of future performance. These tools may also be used to validate requested procedures and flight plans to ensure that no constraints (e. g., thermal or data limits) are breached.

Customer API. Through the customer API, the mission operator has full control of the mission. There are two main API endpoints: the *mission planning* interface and the *telemetry & payload data* interface. The former can be used to control the various payloads on the satellite via external planning software, while the latter allows the customer to oversee the mission by providing access to downloaded telemetry data as well as collected payload data, e. g., images taken by a camera payload. Operation requests are represented as generic lists of time periods indicating when each payload module should be activated. This allows the mission operator to focus on his interests (e. g., with respect to certain areas of interest where Earth observations should be acquired) and compile a matching schedule without the need to think about required attitude control or preheat constraints. This step is handled by the mission planner. The plan resolver converts the scheduled payload activation intervals to a flight plan that can be uploaded to the satellite, thereby adding the necessary slewing or preheating maneuvers. After the feasibility of the resulting plan is ensured and potential conflicts are resolved, the planner finds the next ground station pass in which the plan can be uploaded to the satellite and triggers an upload request to the GOMSPACE ground station network (GSN).

The mission planning API and the telemetry & payload data API are the natural interfaces for integration with the LEOPOWVER toolchain described below.

LEOPOWVER INTELLIGENCE

This section presents the central features and components of the LEOPOWVER software infrastructure for safe, fully automated, energy-efficient, and intelligent operation of LEO satellites and satellite constellations.

At its core, LEOPOWVER takes over the task of continuously maximizing payload utilization while eliminating the risk of overstraining the power budget

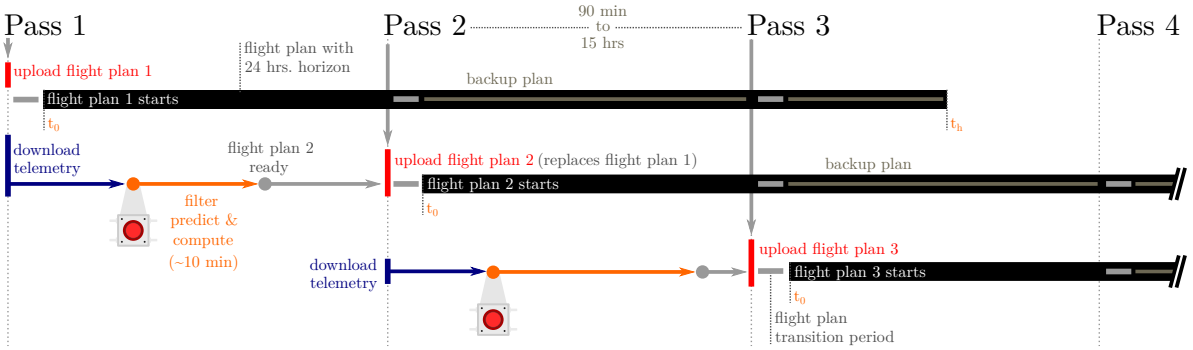

Figure 2: Receding-horizon scheduling along different passes over the ground station. The computed flight plan extends well beyond the needed time horizon to serve as a backup plan in case that the next update fails.

at any moment. The approach is flexible in the way that it can express the intentions of spacecraft engineers and mission operators with respect to the finer optimization goals.

While running, LEOPOWVER continually triggers a sophisticated decision-making procedure to deliver optimal schedules ready to be uploaded before each ground station pass. Although the tool is able to work autonomously, a human operator can be kept in the loop to oversee and safeguard the entire operation. An *orchestrator* acts as the central coordinator of the automated process within the toolchain involving mission models, deep battery models, and a scheduler engine as described below.

The overall process is a resourceful variant of receding-horizon scheduling, a well-known approach[10] to perpetuated finite-horizon scheduling. Our instantiation of the receding-horizon principle is visualized in Figure 2. In a nutshell, the tool can be configured to calculate schedules over some predefined time interval I in the order of one or two days. These schedules are continually re-computed based on the latest available information, i. e., telemetry. Whenever the satellite passes over the ground station, the most recent schedule is offered for upload. Owing to the LEO context, the time span between any two consecutive passes lies somewhere between 90 minutes and 15 hours.

Figure 3 shows a schematic overview of all components the LEOPOWVER toolchain consists of.

Orchestrator. The Orchestrator takes care of coordinating the availability of up-to-date information within the toolchain. In particular, it interfaces with the satellite operator's API in order to receive the latest telemetry of the satellite. The telemetry, in particular logs of voltage and current measurements of the satellite's battery, is fed to the Deep Battery model. And the flight plans provided by the DP module are transferred into the required format and delivered to the satellite operator's API, after being checked for plausibility and feasibility.

DP Scheduling Engine. The dynamic programming (DP) scheduling engine is the central algorithmic component of LEOPOWVER. It receives the access windows regarding the different payloads on board, together with the background load and state of charge estimates regarding the battery at t_0 (the beginning of the scheduling interval I). The DP delivers a set of tasks to schedule that is optimal with respect to the stated objectives, ensuring that the battery stays above a minimum state of charge. A natural (yet simple) example for such an objective would be to assign a certain reward to each executed task and then to maximize the total reward accumulated. Notably, the objective is highly customizable and easy to adapt to the mission operator's needs.

Dynamic programming[11] proved to be a very efficient way to solve these kinds of optimization problems. Partial solutions that potentially lead to optimal solutions are stored in an efficient data structure called the *DP table*. The existing entries are then used to further explore the options and to gradually fill the table. Efficiency directly correlates with the size of the DP table, i. e., the number of entries. Therefore, highly sophisticated antichain-based pruning methods are implemented to remove unnecessary entries and exclude them from further analysis.[12] For maximum performance, this part of the toolchain is implemented in Rust*.

*`https://www.rust-lang.org/`

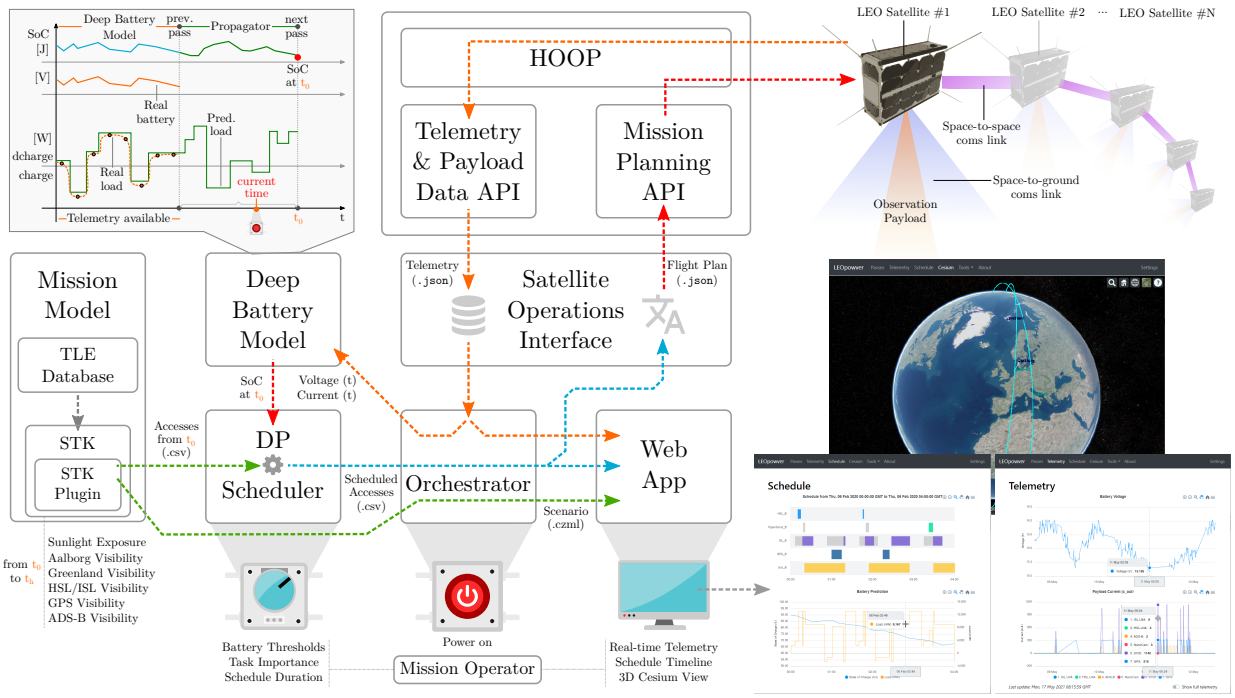

Figure 3: The LEOPOWVER toolchain interfacing with HOOP for the operation of LEO constellations.

Deep Battery Model. One of the innovative contributions to the field of power-aware scheduling is the integration of the Kinetic Battery Model (KiBaM) in the workflow of cost-optimal task scheduling. The KiBaM stores its charge in two parts, namely the available charge $a(t)$ and the bound charge $b(t)$. When a load $\ell(t)$ is applied on the battery, only the available charge is consumed instantly, while the bound charge is slowly converting into available charge via diffusion, as such representing chemically bound energy inside the battery. Diffusion can also happen in the other direction, depending on the amount of both types of charges. One can think of the KiBaM as two wells holding fluid, interconnected by a small pipe. A non-negative diffusion rate v controls the diffusion speed. The two-dimensional state of charge (SoC) of the KiBaM is mathematically evolving according to the two coupled differential equations $\dot{a}(t) = -\ell(t) + v \cdot \big(b(t) - a(t)\big)$ and $\dot{b}(t) = v \cdot \big(a(t) - b(t)\big)$, assuming that both wells have the same capacity. Unlike linear battery models, the KiBaM captures a number of non-linear effects of real batteries, like the recovery effect and the rate-capacity effect.[13]

Within LEOPOWVER, the battery model is able to learn and adapt based on the latest in-orbit measurements, as displayed in the upper left of Figure 3. To this end, the SoC of the battery at the beginning of the scheduling period is obtained. The Deep Battery model exploits the available telemetry of the satellite to properly estimate the real load the battery was feeding energy to and the battery voltage that is used to learn the estimated SoC during the period. While earlier work has used a relatively simple Kalman filter,[4,12] we are now using more advanced intelligence for this purpose. Both battery voltage and current telemetry are sampled on the satellite, which together can be used to determine the output and input load of the battery at least once every two minutes (apart from gaps in the downlinked data due to other storage and transmission priorities). By means of current integration (a. k. a. Coulomb counting[2]), the evolution of the stored energy can be properly approximated, and the actual SoC at the end of the telemetry period can be determined while learning adjustments to the battery model parameters via advanced learning techniques. Next, the propagator operates for the period when no telemetry is available and propagates the battery SoC using the predicted (scheduled) load and the kinetic battery model. At the end of the process, the SoC at time t_0 is made available to the DP scheduler.

Mission Model. The LEOpowver intelligence relies on accurate mission abstractions for which STK[14] is leveraged. To this end, a dedicated STK plugin interfaces with STK core functions to access state-of-the-art orbital propagators, advanced sensor modeling, and access calculations.

The module can interface with a database to obtain the latest orbital parameters of the satellites, as well as precise ground station location, altitude, and elevation profiles. In this, the Two-Line Element (TLE) format, which encodes in two lines of text all the necessary Keplerian parameters defining the orbit of the satellites, is leveraged. Afterwards, STK's built-in Simplified General Perturbations 4 (SGP4)[15] algorithm is used to propagate the trajectories of the satellites into the future. SGP4 is based on accurate analytical and numerical methods considering perturbations caused by the Earth's shape, drag, radiation, and gravitational effects from other bodies such as the Sun and Moon. If needed, the LEOpowver toolchain is prepared to work with more precise propagators such as HPOP[16].

The resulting trajectories are enriched by sensors mimicking the antenna and the coverage of each payload in the evaluated mission (missions with more complex RF payloads can also profit from the Communications module of STK). Next, we capture the contact windows between ISL sensors among flight segments, as well as space to ground segment contacts. To this end, the rich visibility constraints interface provided by STK is leveraged, e. g., ground station elevation angle profiles, maximum link distance, and sun angle exclusion. On the ground part of the model, the territories of interest are modeled by polygons, and the ground stations of the mission are placed at their actual site. We also care about sunlight access for each of the satellites in order to model the time episodes in which the on-board batteries are charged. Sunlight exposure is computed internally in STK using accurate planetary dynamics and exported to the DP engine in the form of simple tables indicating the start and end time of each exposure episode.

Based on this STK scenario, the plugin is able to compute relevant accesses, which are delivered to the DP scheduler in `.csv` format. When needed, obtained windows can be partitioned to allow the DP module to take granular scheduling decisions. Furthermore, the STK script also exports the resulting scenario as a `.czml` file. This file is then fed to a Web App that uses the CesiumJS library* to visualize an interactive 3D view of the scenario accessible via any modern web browser. The whole scenario can be intuitively explored by manipulating the simulation time using the playback and time control features. All of the described processes are fully automated and require no human intervention whatsoever.

*`https://cesium.com/platform/cesiumjs/`

Satellite Operation Interface. This interface bridges the gap between real satellites and the LEOpowver approach to dynamic programming and learning. It serves two main purposes: First, this module collects telemetry measurements and stores them in an internal database where the Deep Battery model can access the data. And second, it sends computed flight plans to the satellite operator's API, thereby converting the abstract schedule representation into the desired output format that can be uploaded to the satellite. Since these are the only two interfaces to external systems, an integration of the LEOpowver tool with existing satellite operation frameworks like HOOP is fairly easy.

GomX–4 MISSION OVERVIEW

Before providing details regarding the in-orbit test campaign carried out with the combination of HOOP and LEOpowver, we provide an overview of the GomX–4 mission and its objectives.

The GomX–4 program is a research and development mission led by GOMspace in partnership with the Danish Defense Acquisition and Logistics Organization (DALO), the Technical University of Denmark, and the European Space Agency. The mission consists of two 6U CubeSats, GomX–4A and GomX–4B, both of which carry a number of payloads. These payloads demonstrate key enabling technologies for future nanosatellite constellations, namely orbit maintenance, inter-satellite communication, high-speed downlinking, and advanced remote sensing.[17]

The two satellites were launched into a 500 km altitude Sun-Synchronous Orbit (SSO) on February 2, 2018, from the Jiuquan Satellite Launch Center in China. Following six weeks of commissioning activities, the satellites entered a six-month technology demonstration phase, during which the payload and operations concept was demonstrated. The satellites have an expected lifetime of 3 to 5 years.

Inter-satellite communication is considered a key enabling technology for future nanosatellite constellations. The GomX–4 mission demonstrated the functioning of an S-band link between the GomX–4A and GomX–4B satellites. To allow the link to be

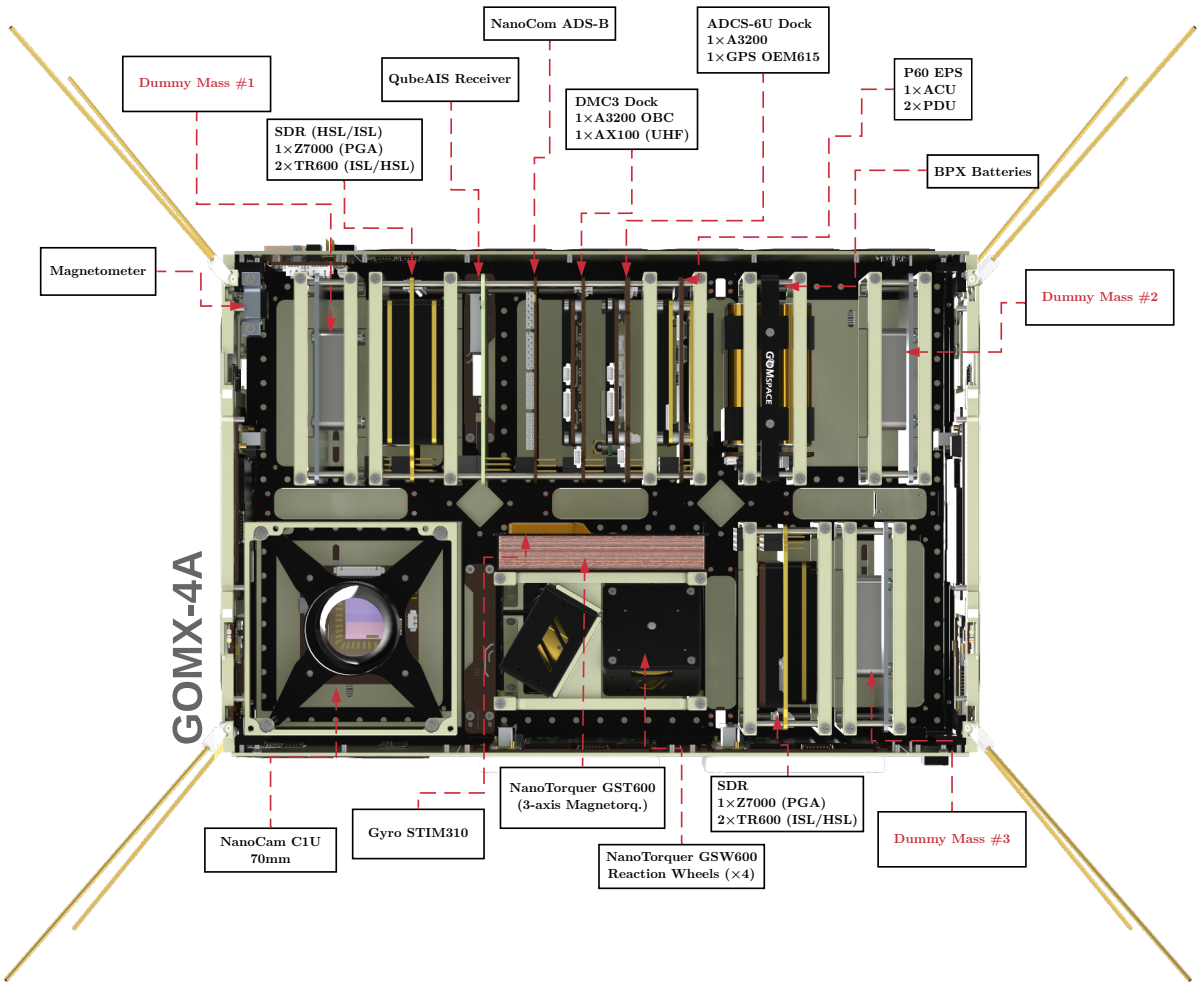

Figure 4: Internal layout of GomX–4A.

established, an inter-satellite separation of less than 4500 km must be maintained. To ensure this, the GomX–4B satellite is fitted with a cold gas propulsion system. No propulsion system is present on the GomX–4A satellite. The inter-satellite link was demonstrated during the technology demonstration phase.

The concrete application context of the GomX–4A mission is the surveillance and collection of remote sensing data of the Arctic region, especially the Greenland territory. For this purpose, the satellite is fitted with an imaging device and with an ADS-B receiver for aircraft tracking. The satellite is also known under the name Ulloriaq, the Greenlandic word for "star".

Both satellites are commanded from a ground station located in Aalborg, Denmark (57°1′22″N, 9°58′41″E). Two antennas are available at this ground station, one in UHF and one in S-band. The UHF antenna is used for robust low-data rate tasks such as telecommands and the reception of telemetry. The S-band antenna is used to download payload data. Due to the orbit properties of the SSO, each satellite is visible to the ground station five to six times per day. These contact opportunities are clustered, typically in the early morning and late afternoon each day.

Platform

The platform of the GomX–4 satellite (see Figure 4) maintains the stability and functioning of the satellite. The satellite follows the standard GomSpace 6U architecture, i. e., a decentralized architecture where the failure of any one node does not impact the functioning of any other.

- **EPS.** The Electrical Power System consists of a NanoPower P60. This system manages input power from the solar panels fixed to the outside of the satellite and supplies power to the other

components on board. Power is stored using a BPX battery subsystem.

- **Solar Panels.** GomX–4 uses solar panels to obtain electrical power. These panels are positioned on one side of the satellite, meaning the satellite must be orientated correctly to achieve maximum charging.
- **ADCS.** The ADCS subsystem determines and controls the satellite attitude. The system on GomX–4 uses sun sensors, magnetometers, and magnetorquers to perform this function.
- **OBC.** The On-Board Computer (OBC) provides processing and data storage capabilities. Flight plans activated on the OBC may instruct other components on the satellite according to scheduled commands.
- **UHF Link.** The UHF Link enables communication between GomX–4 and the ground station at Aalborg. Telemetry and telecommands (i. e., flight plans) are transferred via this link.
- **High Speed Link.** The High Speed Link (HSL) uses an S-band radio link to provide a fast data connection to the ground station at Aalborg, used to transfer acquired payload data to the ground segment.

Payloads

The payload of the GomX–4A satellite is composed of a number of sensors designed for communication and surveillance.

- **NanoCam Camera.** The NanoCam C1U Camera is an RGB imaging device that is used to observe the Greenland territory.
- **Inter-Satellite Link.** The Inter-Satellite Link (ISL) enables data transfer between the GomX–4 twins and other LEO satellites close enough (within 4500 km) that are able to communicate in S-band.
- **Global Positioning System.** The new Novatel OEM719 payload is an additional Global Positioning System (GPS) receiver that will eventually replace the Novatel OEM615 also on-board the satellite. It enables GomX–4A to receive GPS satellite location signals to determine its precise local position.
- **ADS-B Receiver.** The Automatic Dependent Surveillance–Broadcast is a surveillance technology allowing the tracking of the position, velocity, and other sensor data broadcast by aircraft. It is used by GomX–4A to track aircraft over the Atlantic where ground-based reception is difficult.

UHF is enabled whenever Aalborg is in line-of-sight, so it can be considered as a recurring background load on the battery, similar to sunlight exposure (but with opposite effects on the battery). ISL transceivers are installed on board for gaining experience with data transfer between satellites, whenever possible over the poles. The restriction to over-the-pole episodes is rooted in the necessity to comply with international radio frequency regulations regarding the use of S-band over inhabited regions.

EXPERIMENTAL RESULTS

This section presents the empirical evaluation results of a thorough test campaign where GomX–4A has been operated by the Powver-HOOP toolchain. During a period spanning from April 12 to May 14, a variety of different in-depth experiments were carried out in orbit.

Scenario Set-Up

The concrete mission scenario we focus on in detail consists of the GomX–4A satellite and an imaginary satellite GomX–4C acting as a receiver/transmitter of data in the constellation. The latter is placed in orbit in such a way that the resulting communication opportunities are non-trivial to plan for. For this, the STK plugin computes the corresponding TLE of the imaginary GomX–4C satellite, based on the actual data for GomX–4A with RAAN angle and mean anomaly shifted by 10°. The RAAN angle indicates the angle of the orbital plane in the equator, while the mean anomaly stands for the position of the satellite along the trajectory in the orbit.

ISL tasks are allowed to occur only when the ISL sensor on GomX–4A can point to GomX–4C on a distance no larger than 1300 km. As illustrated in Figure 5, the resulting configuration renders two satellites that separate maximally at the equator and become aligned in an along-track configuration as they come close over the North and South poles. Exactly in this condition, the inter-satellite distance is minimal, and the inter-satellite link antennas become aligned. As a result, ISL transmission between GomX–4A and GomX–4C is only possible during over-the-pole episodes. Thereby, potential interference with ground radio-frequency services operating on S-band in populated regions is avoided. Depending on the small orbital variations, these episodes last

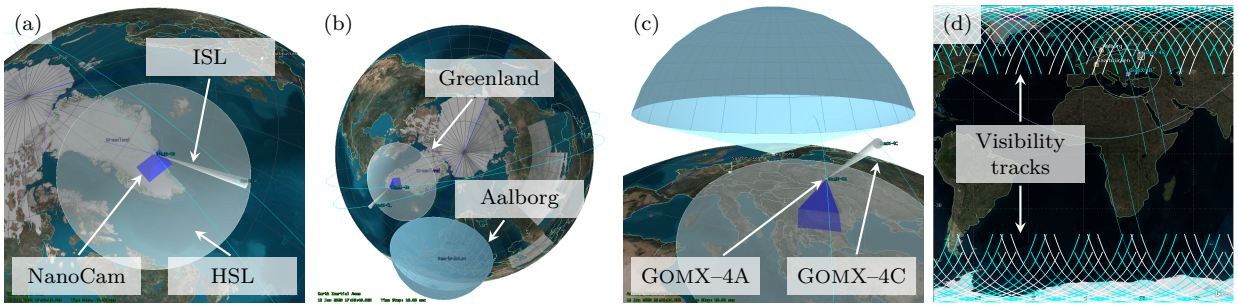

Figure 5: Different GOMSPACE mission scenario situations visualized in STK.

around 20 minutes. Since the power demand of the ISL payload is considerable, we allow for intermittent utilization of the resource. To account for this, the ISL windows are partitioned into three smaller chunks of 7 minutes each.

Flight plans are usually uploaded via the UHF link during passes over the ground station in Aalborg. Since the process of swapping the flight plan needs exclusive access to the satellite, and to make the scenario more interesting, we introduce two different ground stations to the scenario that are used to downlink collected data via HSL: Teófilo Tabanera Space Center in Córdoba, Argentina (31°31′30″S, 64°27′46″W) and Svalbard Satellittstasjon in Svalbard, Norway (78°13′47″N, 15°24′28″E). According to GOMX–4A specifications, a conic HSL sensor is configured with 2000 km maximum range and 65° of half-angle pointing in the nadir direction (i. e., to ground) while the HSL downlink antennas of the two ground stations in Córdoba and Svalbard are modeled each by a sensor with a half-angle of 70° and maximum range of 2000 km.

The NanoCam is modeled by a rectangular sensor pointing in nadir direction with 16° and 23° of across and along angles, respectively. Visibility conditions between the NanoCam and the Greenland territory—that is modeled by a polygon—are stored as potential camera tasks.

Finally, GOMX–4A is configured to use its ADS-B receiver to track aircraft only when orbiting over the Atlantic Ocean, where coverage with terrestrial antennas is difficult. While the Atlantic is again modeled by a polygon, the ADS-B receiver has no attitude constraints of interest and uses a simple line of sight constraint.

To comply with radio frequency regulations, no real data other than the satellite's telemetry was transmitted using HSL and ISL during the experiments. This was realized on the satellite by configuring the radio module to not use the power amplifier (PA) connected to the antenna. To compensate for the energy consumption of the PA, the GPS payload was simultaneously activated during HSL and ISL tasks.

Table 1: Experiment Summary.

Start of Schedule	Max El.	Status
May 10, 00:57	85.85°	✅ Executed
May 10, 13:50	36.22°	✅ Executed
May 10, 15:23	28.42°	✅ Executed
May 11, 00:34	53.59°	✅ Executed
May 11, 15:01	44.67°	⚠ Backup Plan

✅ – flight plan successfully uploaded
⚠ – upload failed, but earlier plan continued

The scenario can be easily chained up to arrive at larger satellite constellations with non-trivial communication opportunities. Since future GOMSPACE missions will exploit ISLs for continuous networked operation, interdependencies in data flow models are appealing extensions of the present work, where scalability needs to be mastered in terms of satellite fleet and orbital parameter diversity.[7]

Evaluation

The entire time of exclusive experimental access to GOMX–4 spanned a period of 758 consecutive hours in which GOMX–4A was operated by the POWVERHOOP toolchain. In the following, we are focusing on two days from our experimental case study. Starting on May 10, 2021, at 00:57, five Aalborg ground station passes were targeted for the consecutive upload of flight plans. To obtain a reliable uplink, only passes with a maximum elevation higher than 25° were considered. This means that five schedules were made available in total, each having a scheduling horizon of 24 hours. The precise timings when each schedule became valid are listed in Table 1.

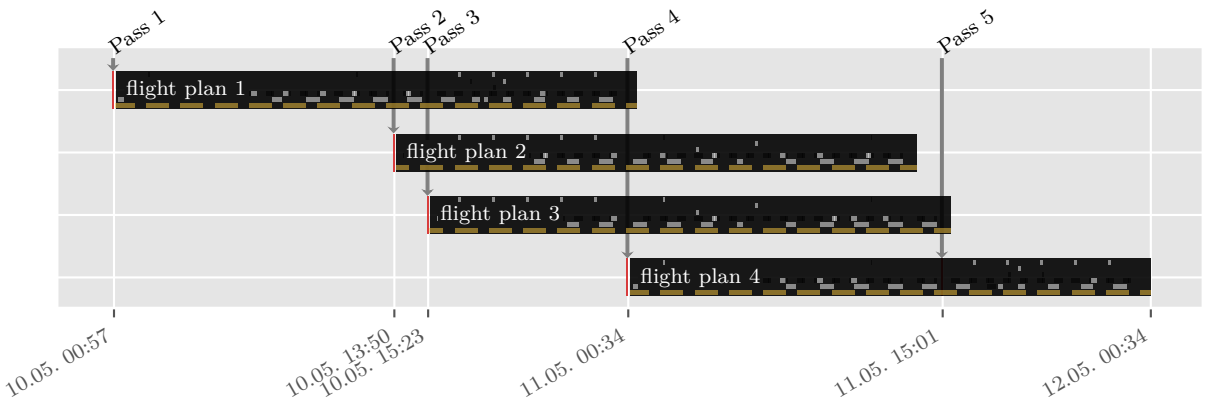

Figure 6: Experiment overview with the four uploaded flight plans.

LEOPOWVER in Action. Out of the five flight plans, four were uploaded successfully to the satellite. The last plan suffered from transmission issues on the ground segment and could not effectuate on the satellite. Even with the best precautions, such radio communication failures can occur anytime in ground-to-satellite communication. This stresses the importance to compute schedules that cover well beyond the next ground station passes. In fact, this failure had little impact on the overall mission since the previously valid schedule simply continued to execute. Although this may in general lead to a substantial drift between the predicted battery behavior and real battery voltage, the Deep Battery model takes care of that and corrects the predictions over time when new telemetry measurements arrive.

A temporal overview of the full experiment horizon with the four uploaded flight plans is shown in Figure 6. Each time the satellite passes over the ground station and a fresh and improved plan is uploaded to the satellite, the remaining parts of the previous plan are overwritten. Figure 7 illustrates this behavior for the whole experiment. At the top, all possible payload utilization windows and sunlight exposure episodes are shown. Access windows that were actually scheduled by our tool are marked in blue, or gray if skipped. The three green lines mark the positions where new plans were uploaded (and the dashed red line where the upload failed).

Predictions vs. Reality. The middle and bottom of the figure show the predicted evolution of the battery with respect to the SoC of the KiBaM model and the load profile that is applied to the battery. According to telemetry measurements, the initial SoC was assumed to be around 75 %, and the model predicted the SoC to decrease to 63 % at the moment the second flight plan should take over. However, this was not the case either because the true initial SoC was actually higher than expected or because the model was too pessimistic concerning the payload energy utilization. As a result, the initial SoC for the second plan was corrected upwards by the Deep Battery model by around 5 %, allowing the second flight plan to perform more activities, hence increasing the satellite's productivity. For the next pass 90 minutes later, a third flight plan was computed. This time, the Deep Battery model induced no correction in either direction, meaning that the plan used the predicted model estimates unaltered to derive the initial SoC. Similar to the second plan, the initial SoC of the fourth plan was again corrected upwards by the Deep Battery model.

The battery voltage measurements that were downlinked from the satellite and used in the Deep Battery model updates are shown in Figure 8. The blue line again represents the evolution of the available charge of the KiBaM model. The individual telemetry data points are depicted as purple hexagons, connected by an orange line that linearly interpolates the given discrete set of measurements recorded. The battery pack on board GOMX–4A has a maximal voltage of 16.2 V. The lowest allowable operational limit for the battery is 14.8 V. If the voltage falls below this nominal voltage limit, a safe-mode is triggered on the satellite that disables nonessential payloads until the battery has recovered. According to the engineers, a battery pack with 14.8 V contains roughly 55 % of energy. Therefore, in the plot, a voltage of 14.8 V is aligned with 55 %.

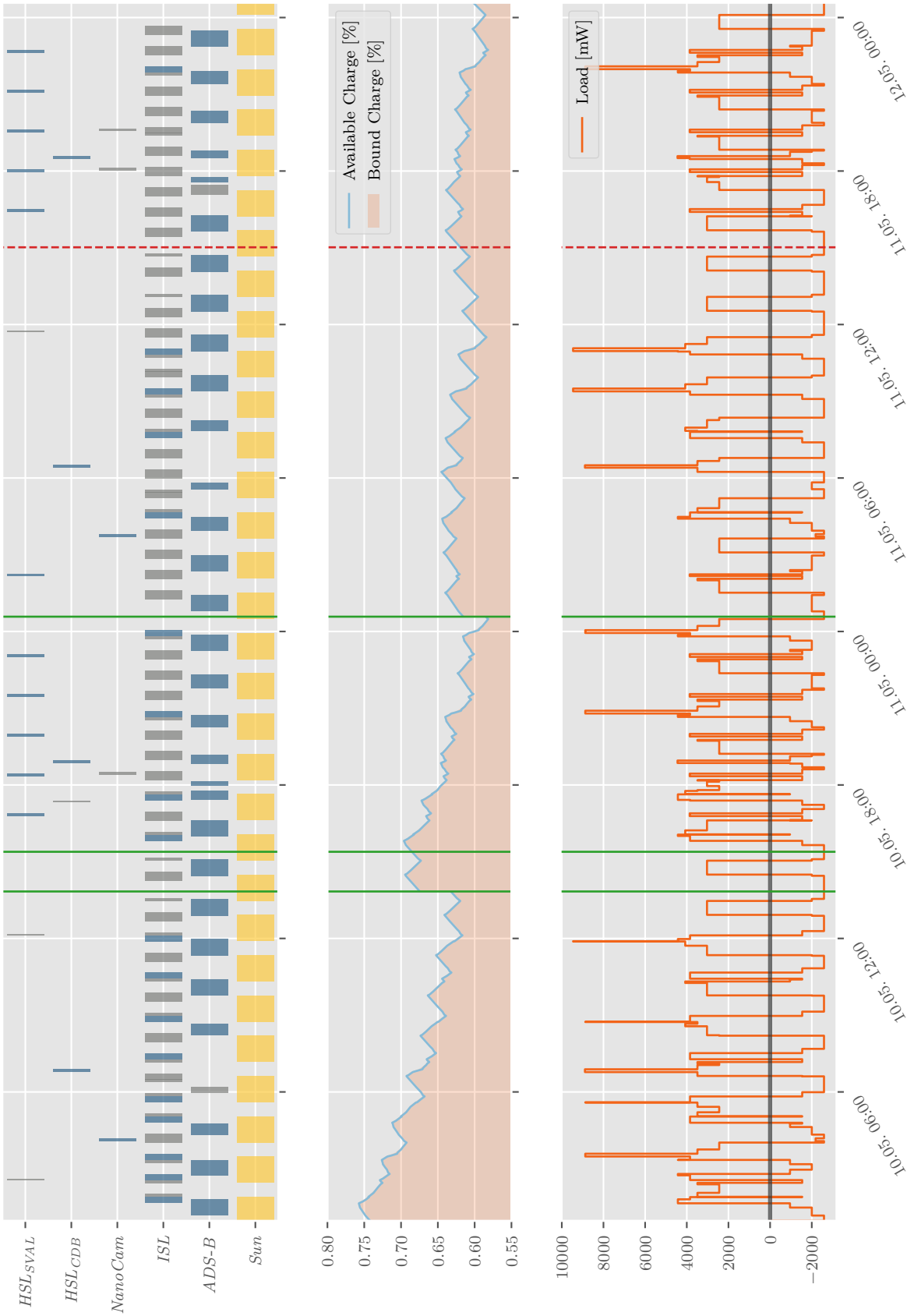

Figure 7: Tasks and predicted battery behavior of the receding-horizon schedules.

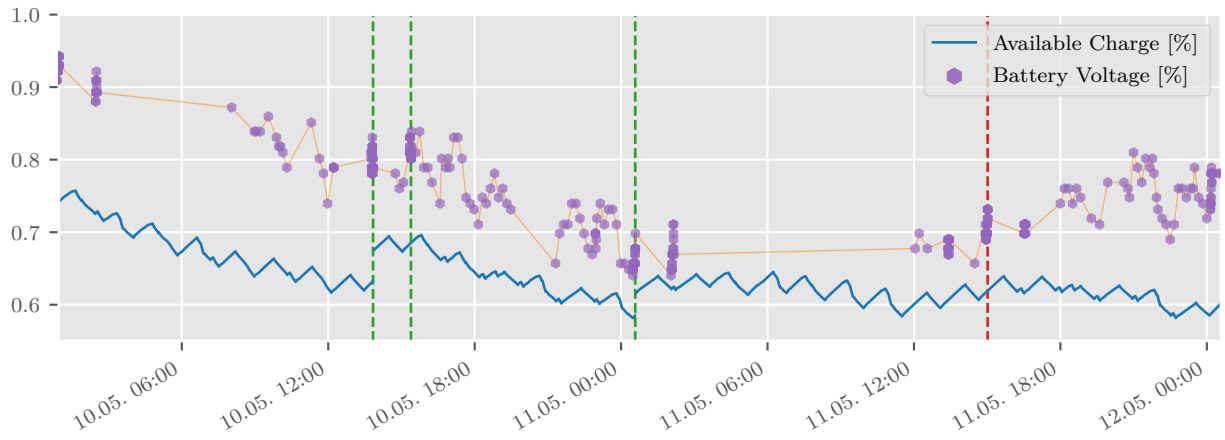

Figure 8: GomX–4A battery voltage telemetry during the experiment.

Effect of Receding-Horizon. For the purpose of analyzing the overall effect of the receding-horizon approach, a monolithic scheduling alternative was computed, using the same initial scenario setting as the real experiment but long enough to cover the entire experiment horizon in one shot, thus without considering any feedback from telemetry received during the experiment. The receding-horizon schedules and the monolithic schedule are superimposed in Figure 9. The payload utilization windows in the upper half of the figure are now split into two parts, where the top part belongs to the receding-horizon schedule, and the bottom part belongs to the single schedule. Payload windows that are chosen by the tool to be executed only in the receding-horizon version but not in the single schedule are highlighted in green, while windows that are only scheduled in the monolithic schedule are colored red.

The computed schedules differ in many places, but if one looks at the total task counts being scheduled, the quantitative difference is that the receding-horizon schedule can sustain one more HSL task and six more ISL chunks than the single monolithic at the price of waiving two ADS-B opportunities. Note, however, that from an energy budget perspective, HSL and ISL are much more power-demanding than ADS-B (around one order of magnitude). These findings make sense since the Deep Battery model factually added energy to the system twice in the receding-horizon setting, based on the telemetry received. This demonstrates the importance of automated model learning techniques whenever working with battery models based on some theoretical assumptions about the consumers.

FUTURE DIRECTIONS

This section provides a survey of lessons learned and further steps to be taken.

HOOP and the GomX Series. The future direction as per GOMspace roadmaps are:

- **GomX–5.** GOMspace is currently developing the next satellite in the GomX series, GomX–5. This mission, developed with the European Space Agency, is expected to launch in 2022 and will be fully operated through the HOOP platform, taking advantage of automation to optimize operations. GomX–5 will demonstrate new capabilities in space, including increases in payload downlink volume, maneuverability for changes in orbit, and satellite position accuracy. The satellite will be equipped with advanced payloads and will carry out technology demonstration missions.
- **HOOP.** The development of HOOP is continuing, most recently under the MCOP Advanced project funded by the Luxembourg Space Agency and supported by the European Space Agency. This development includes advanced simulation capabilities, including enhanced flight dynamics tools and features to enable the management of larger constellations. HOOP will be deployed for an increasing number of commercial and scientific satellite missions, its automation capabilities enabling the operation of a growing number of smallsat constellations.

LEOpowver. The following technical challenges are examples for the next steps that are planned to

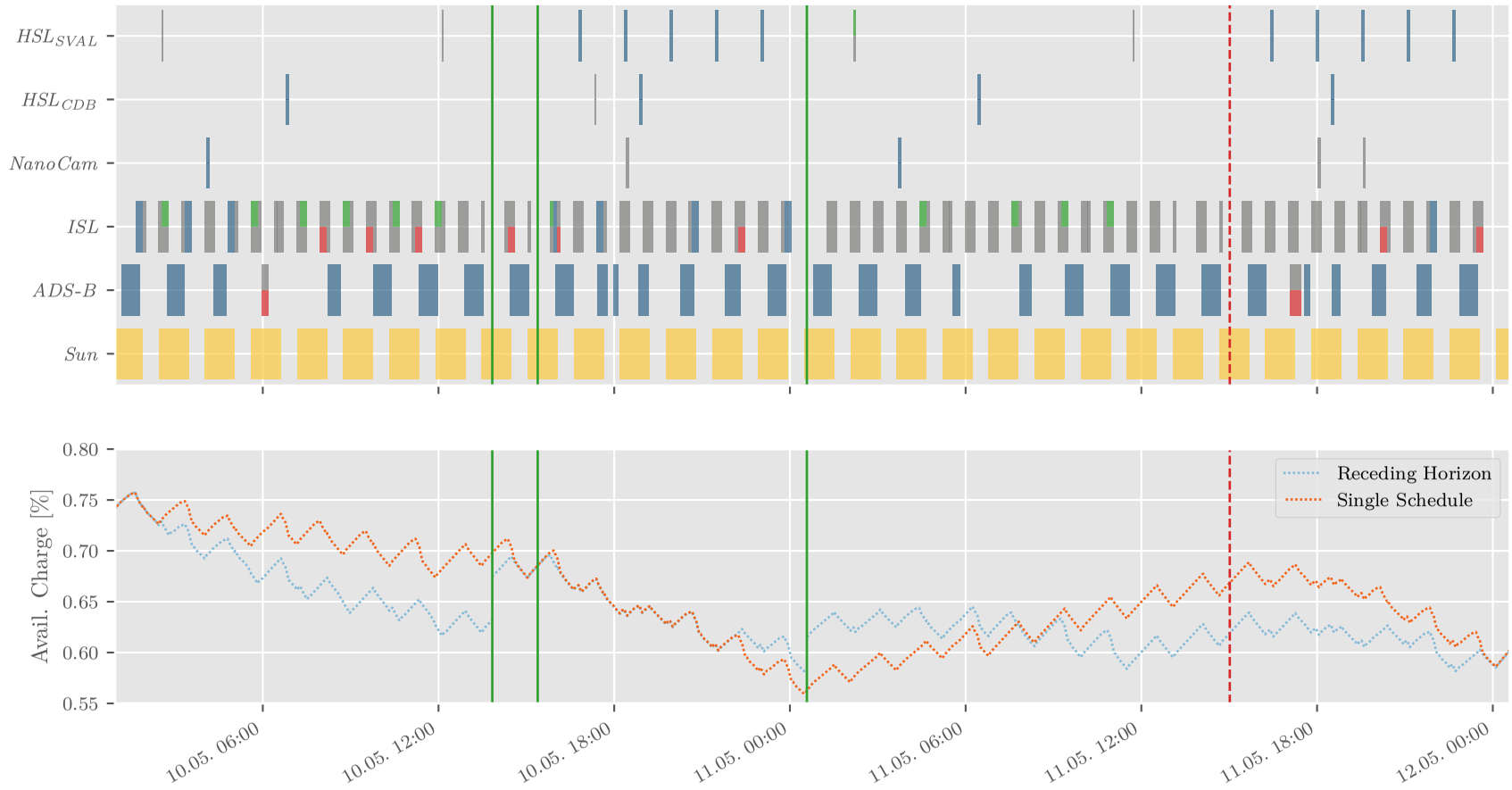

Figure 9: Comparison of the receding-horizon schedules with a monolithic schedule covering the entire experiment horizon.

be tackled by the LEOPOWVER ambitions. They will be embedded into user-centric activities especially targeting start-ups and business customers.

- **Data Flow Modeling.** The accurate commanding of space-to-ground and space-to-space communication subsystems depends on the information that is to be transferred via such links. The data volume to be transported is, in most cases, well known in advance as it is either generated periodically (telemetry) or triggered by commands sent in advance (science or mission data). In this context, well-known multi-commodity flow models and algorithms can be leveraged.[18] Indeed, we are putting the technology in place for LEOPOWVER to account for accurate data flow models to ensure more tight control of the utilization of power-hungry communication resources.[6]

- **Expression of Mission Objectives.** The current toolchain provides the mission operator with convenient options to define task importances and thus to guide the automated operations. However, complex constellation operations will require more powerful semantics to characterize specific mission goals. For example, tasks in constellations might be satellite-agnostic in the sense that acquisitions over a specific region can be executed by any of the satellites in the constellation. Furthermore, task dependencies are a topic, e. g., if some data acquisition task is scheduled, then some downlink task should follow in the near future. The configuration of these constraints is possible already but can become more flexible, and indeed we are about to release an expressive language for tasking constellations. Algorithmically, these aspects can be framed as variants of the more general assignment problem, for which powerful algorithmic approaches are readily at hand at Saarland University to boost the capabilities of LEOPOWVER.

- **Battery Degradation.** Batteries are known to degrade over time, meaning that they lose more and more storage capacity. While battery degradation cannot be avoided, being able to accurately predict and model it is crucial to increase the operational lifetime of the mission. The LEOPOWVER Deep Battery model will be extended to learn on-the-fly when the battery behavior changes and thus predict the battery end-of-life performance.[19] This important phenomenon can then be considered during the scheduling phase, e. g., so that the affected satellite is scheduled to perform on nonessential tasks. When used in constellations of nodes with similar battery models, the learning could further profit from multiple data points.

In fact, these lists of action points appear a lot more separate than they actually are. The experiments and the collaboration between the teams at GOMSPACE and Saarland University have been so successful that the technical and algorithmic progress made on the LEOPOWVER side likely will go hand-in-hand with the advancements planned for HOOP.

CONCLUSION

Driven by recent breakthroughs in technology, the interest of the space community in LEO satellites has reached an exorbitant level. Thousands of scheduled launches are planned for years to come, opening the door for a sharply increasing market. The skyrocketing number of satellites in orbit calls for the need for automatic satellite operation and management solutions.

GOMSPACE and Saarland University have successfully cooperated to combine their respective expertises: GOMSPACE as an industrial market leader in the commercialization of nanosatellites and Saarland University as distinguished technology provider regarding ultra-effective energy-aware cost-optimal task scheduling. We have presented POWVER-HOOP, a fully automated satellite constellation operation framework, exploiting optimal computing techniques such as dynamic programming and self-learning models. The dual-satellite GOMX–4 mission by GOMSPACE was used to validate and demonstrate the approach in orbit, allowing our integrated software to be distinguished as *orbit-proof*. The LEOPOWVER software is unique in its features:

- The Deep Battery model exploits the non-linear dynamics of the kinetic battery model and is self-adjusting continuously with the latest telemetry received from orbit.

- The scheduling is perpetuated by a receding-horizon strategy.

- Data transfer between different satellites in orbit is accounted for by providing support for the scheduling of inter-satellite links.

- The core algorithmic problem is framed as dynamic programming with antichain-based pruning.

- The resulting software framework is directly ap-

plied for the automated management and operation of the GomX–4 LEO mission.

- The approach is carefully crafted to take into account the usability by the space engineers and robustness against failures of parts of the toolchain.
- An extensive in-orbit test campaign validates accuracy, efficiency, scalability, and robustness with respect to the operational requirements and constraints of LEO constellations.
- The software architecture is designed in such a way that it can flexibly be embedded into the entire range of applications in LEO (and beyond). Connecting it to HOOP has been very straightforward and successful.

Further improvements and enhancements are on the roadmap. The resulting machine learning approach from LEOpowver as well as the operation scope of HOOP are highly flexible and can be scaled to more complex networked constellation-class missions. In this view, more elaborated data flow and battery models and more powerful mission semantics are being integrated into LEOpowver and thus made available to operators of larger missions via the enhanced HOOP interface.

Acknowledgments

This research has received support by the ERC Advanced Grant 695614 (powver), the ERC Proof of Concept Grant 966770 (LEOpowver), and by the DFG Grant 389792660, as part of TRR 248 (CPEC).

References

[1] Hannah R. Goldberg et al. "The Juventas CubeSat in Support of ESA's Hera Mission to the Asteroid Didymos". In: *Proceedings of the AIAA/USU Conference on Small Satellites*. Instruments / Science, SSC19-WKIV-05. 2019. URL: https://digitalcommons.usu.edu/smallsat/2019/all2019/73/.

[2] Holger Hermanns, Jan Krčál, and Gilles Nies. "How Is Your Satellite Doing? Battery Kinetics with Recharging and Uncertainty". In: *Leibniz Trans. Embed. Syst.* 4.1 (2017), 04:1–04:28. DOI: 10.4230/LITES-v004-i001-a004.

[3] Gilles Nies et al. "Mastering operational limitations of LEO satellites – The GomX-3 approach". In: *Acta Astronautica* 151 (2018), pp. 726–735. ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2018.04.040.

[4] Morten Bisgaard et al. "Battery-aware scheduling in low orbit: the GomX-3 case". In: *Formal Aspects Comput.* 31.2 (2019), pp. 261–285. DOI: 10.1007/s00165-018-0458-2.

[5] Juan A. Fraire and Jorge M. Finochietto. "Design challenges in contact plans for disruption-tolerant satellite networks". In: *IEEE Commun. Mag.* 53.5 (2015), pp. 163–169. DOI: 10.1109/MCOM.2015.7105656.

[6] Juan A. Fraire et al. "Battery-Aware Contact Plan Design for LEO Satellite Constellations: The Ulloriaq Case Study". In: *IEEE Trans. Green Commun. Netw.* 4.1 (2020), pp. 236–245. DOI: 10.1109/TGCN.2019.2954166.

[7] Juan A. Fraire et al. "On the scalability of battery-aware contact plan design for LEO satellite constellations". In: *Int. J. Satell. Commun. Netw.* 39.2 (2021), pp. 193–204. DOI: 10.1002/sat.1374.

[8] CS Group. *Orekit*. Version 10.3. URL: https://www.orekit.org/ (visited on 05/18/2021).

[9] NASA Goddard Space Flight Center. *General Mission Analysis Tool (GMAT)*. Version R2020a. URL: http://gmatcentral.org/ (visited on 05/26/2021).

[10] Suresh Chand, Vernon Ning Hsu, and Suresh P. Sethi. "Forecast, Solution, and Rolling Horizons in Operations Management Problems: A Classified Bibliography". In: *Manuf. Serv. Oper. Manag.* 4.1 (2002), pp. 25–43. DOI: 10.1287/msom.4.1.25.287.

[11] Richard E. Bellman. *Dynamic Programming*. 1st ed. Princeton University Press, 1957.

[12] Gregory Stock et al. "Managing Fleets of LEO Satellites: Nonlinear, Optimal, Efficient, Scalable, Usable, and Robust". In: *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 39.11 (2020), pp. 3762–3773. DOI: 10.1109/TCAD.2020.3012751.

[13] Marijn R. Jongerden and Boudewijn R. Haverkort. "Which battery model to use?" In: *IET Softw.* 3.6 (2009), pp. 445–457. DOI: 10.1049/iet-sen.2009.0001.

[14] Analytical Graphics, Inc. *Systems Tool Kit (STK)*. Version 12.2. URL: https://www.agi.com/products/stk/ (visited on 05/18/2021).

[15] David A. Vallado et al. "Revisiting Spacetrack Report #3". In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. 2006. DOI: 10.2514/6.2006-6753.

[16] Ahmed Refaat et al. "High Accuracy Spacecraft Orbit Propagator Validation". In: *Proceedings of the 18th International Conference on Applied Mechanics and Mechanical Engineering*.

Vol. 18. Military Technical College, 2018. DOI: 10.21608/AMME.2018.34732.

[17] Laura León Pérez, Per Koch, and Roger Walker. "GOMX-4 – The Twin European Mission for IOD Purposes". In: *Proceedings of the AIAA/USU Conference on Small Satellites*. Science / Mission Payloads, SSC18-VII-07. 2018. URL: https://digitalcommons.usu.edu/smallsat/2018/all2018/296/.

[18] Juan A. Fraire, Pablo G. Madoery, and Jorge M. Finochietto. "Traffic-aware contact plan design for disruption-tolerant space sensor networks". In: *Ad Hoc Networks* 47 (2016), pp. 41–52. DOI: 10.1016/j.adhoc.2016.04.007.

[19] Erik Ramsgaard Wognsen et al. "A Score Function for Optimizing the Cycle-Life of Battery-Powered Embedded Systems". In: *Formal Modeling and Analysis of Timed Systems - 13th International Conference, FORMATS 2015, Madrid, Spain, September 2-4, 2015, Proceedings*. Ed. by Sriram Sankaranarayanan and Enrico Vicario. Vol. 9268. Lecture Notes in Computer Science. Springer, 2015, pp. 305–320. DOI: 10.1007/978-3-319-22975-1_20.