

## Operational Simulation of Optical Inter-Satellite Links in LEO Constellations

Luca Pizzuto\*

*Galileo Competence Center, German Aerospace Center (DLR),  
Münchener Straße 20, 82234 Weßling, Germany,  
\*Corresponding Author, email: luca.pizzuto@dlr.de*

### Abstract

Optical Inter-Satellite Links (O-ISL) are gaining increased adoption, due to their advantages over standard radio communications in terms of achievable bandwidth and virtually non-existent band interference issues. These benefits are crucial for LEO constellations serving diverse sectors such as disaster management, financial services, law enforcement and intelligence services, which all have a demand for reduced data latency and timeliness of actionable information. For this reason, large constellations foresee the use of or are starting to employ O-ISLs at least for in space communication links. One of the challenges in adopting O-ISLs is the management of the network in space and the data routing within the constellation. This paper focuses on the development of an operational simulator for O-ISLs in LEO constellations using NASA Operational Simulator for Small Satellites (NOS3). The goal is to present the integration and employment of an O-ISL subsystem model into the simulator to examine the operational impact. NOS3 provides a simulation engine for CubeSat and small satellite missions, which integrates the flight software, satellite subsystem models, ground control software and environment simulator to create a realistic platform for testing satellite operations. Multiple instances of NOS3 have been deployed to simulate a constellation and an O-ISL subsystem has been integrated on each satellite of the constellation. A single orbital plane of a Walker Delta constellation consisting of 8U CubeSats is implemented. The O-ISL are modelled as 1U optical heads with a 1-degree field of view and are mounted on the ram and wake direction of the CubeSat, enabling it to establish intra-planar links with adjacent satellites in the forward and backward positions within the same orbital plane. Operating O-ISLs necessitates precise pointing and tracking mechanisms due to the narrow beamwidth of laser communications. Accurate knowledge of satellite position and attitude is critical. Communication establishment combines GPS-derived satellite positions with the propagation of other satellite orbits using periodically updated two-line elements. Once the link is established, multiple operational scenarios are simulated. These include real-time data sharing, where satellites can transfer data, telemetry or commands within the orbital ring. Another scenario involves data relay to ground stations, where the satellite acquiring actionable information over an area of interest but without ground station coverage can relay the data via the optical links within the constellation to a satellite with ground station access, ensuring the data latency is minimized. The simulation environment evaluates the resilience of the intra-planar O-ISL network under dynamic conditions, assessing both the effectiveness of attitude adjustments in maintaining link integrity and the system's ability to meet stringent latency requirements for time-sensitive operations. This paper also discusses the lessons learned from introducing a new subsystem model into the NOS3 engine, the challenges that need to be addressed when doing so, and the fidelity achievable with NOS3 as an operational simulator.

**Keywords:** Optical Inter-Satellite Links, LEO Constellations, NASA Operational Simulator for Small Satellites (NOS3), Operational Satellite Simulation, Satellite Data Routing.

### Acronyms/Abbreviations

Attitude Determination and Control System (ADCS), CCSDS File Delivery Protocol (CFDP), Concept of Operations (ConOps), Earth-Centered Inertial (ECI), Extensible Mark-up Language (XML), Field of Regard (FoR), Free Space Optical Communication (FSOC), General Mission Analysis Tool (GMAT), Global Navigation Satellite System (GNSS), Inter-Satellite Links (ISLs), NASA Operational Simulator for Small Satellites (NOS3), Optical Ground Stations (OGS), Optical Inter-Satellite Links (O-ISL), Orbit Determination and Time Synchronisation (ODTS), Positioning Navigation and Timing (PNT), Precise Orbit Determination (POD), Proportional-Derivative (PD), System Tool Kit (STK), Telecommands (TC), Telemetry (TM), Virtual Machine (VM).

## 1. Introduction

Inter-Satellite Links (ISLs) are communication links that connect satellites directly with each other, rather than relying on ground stations for relaying information. These links enable satellites within a constellation to share data, coordinate operations, and enhance overall network performance by improving constellation coverage and reducing

latency and ground segment dependency, with the result of decreasing operational costs and increasing overall reliability. ISLs are particularly useful for Positioning Navigation and Timing (PNT) constellation concepts since they improve the precision of orbit determination and clock estimation [1][2][3], hence enhancing the positioning performance of the system. For these reasons, multiple Global Navigation Satellite System (GNSS) constellations are already relying on ISLs, like Beidou Navigation Satellite System, and others foresee their adoption for their future generations, like Galileo and GPS [4].

In parallel, Free Space Optical Communication (FSOC) is emerging as a key technology, attracting growing research interest. Compared to traditional radio frequency communication, FSOC offers significant advantages, including higher bandwidth, lower power and mass requirements, enhanced security, and freedom from spectrum licensing constraints [5]. These benefits are particularly valuable for LEO constellations serving time-sensitive applications, such as disaster management, intelligence operations, and Earth observation, where reduced data latency and timely delivery of actionable information are crucial. Given these advantages, the integration of Optical ISLs (O-ISLs) into satellite constellations presents a promising opportunity, combining the benefits of ISLs and FSOC. In addition to enhancing data transfer efficiency, O-ISLs play a critical role in improving PNT services, as they enable more precise orbit determination and clock synchronization compared to radio-frequency ISLs [6].

However, operating a LEO constellation adopting O-ISLs in an efficient and cost-effective way remains to these days a significant challenge. The complexity of space-based network management and data routing is further compounded by the stringent pointing and tracking requirements of optical systems. Additionally, fully optical constellation concepts must contend with atmospheric conditions, such as cloud coverage, which can obstruct optical uplinks and downlinks. Currently, O-ISL planning is conducted conservatively and inefficiently, and no open-source tools exist that can simulate the operational aspects of such constellations with high granularity and detail. To address these challenges and define system-level requirements for future operators, we developed an operational simulator for O-ISLs in LEO constellations by extending an open-source operational simulator from NASA. This work presents the integration of an O-ISL subsystem model into the simulator and examines its application in assessing operational performance and deriving key system requirements.

### *1.1 Overview of Previous Works and Simulation Environments*

Given the growing interest in O-ISLs, various research institutes and optical payload manufacturers have employed a range of simulation tools to define operational and performance requirements and conduct optimisation analyses. In a recent study, Sayyed et al. [7] utilized the OptiSystem [8] software design tool from Optiwave to model the performance of O-ISLs between GEO and LEO satellites under different conditions, including varying inter-satellite distances, atmospheric turbulence, and pointing errors. OptiSystem provides a comprehensive platform for planning, testing, and simulating optical links across a wide range of optical networks at the transmission layer. However, it lacks the capability to simulate the unique conditions of the space environment, including spacecraft dynamics and subsystem interactions. To address this limitation, many researchers in the aerospace domain tend to use the AGI System Tool Kit (STK) [9] software, which facilitates the simulation of constellation networks, orbit propagation and access report generation. For instance, Raffa et al. [10] employed STK to simulate two LEO mega-constellations and derive satellite-to-satellite access reports which were subsequently used to model O-ISLs in Python. Their study focused on optimizing the ground segment by determining the minimum number of Optical Ground Stations (OGS) required to operate the network while ensuring optimal end-to-end access availability for users. Similarly, STK was used in Baltaci et al. [11] to analyse the line-of-sight dynamics between satellites in a LEO constellation modelled after Starlink. The study examined O-ISLs in both the North-South direction, meaning between satellites within the same orbital plane, and the East-West direction, meaning between satellites in different, crossing orbital planes. The findings highlighted the challenges of maintaining interplanar O-ISLs due to frequent link fluctuations and dynamic handover conditions, both of which impose significant constraints on the design of optical terminals and the management of in-orbit network routing. These studies demonstrate STK's strength in conducting line-of-sight analyses and generating access reports, along with its ease of use and high simulation speed. Another commonly used simulation tool is NASA's General Mission Analysis Tool (GMAT) [12], an open source software primarily used for mission analysis and optimisation, with a focus on trajectory design, orbit determination, and mission planning. In Kling et al. [13], GMAT was utilized for a trade-off study comparing the impact of ISLs, edge computing, and a ground station network on data latency reduction in various LEO constellations. The study concluded that ISLs offer the most effective solution for minimizing data latency while also reducing ground segment requirements and operational costs. In this context, GMAT was particularly useful for modelling constellation dynamics, with ISLs represented in terms of line-of-sight, achieving a level of detail comparable to that provided by STK.

## 1.2 Aims & Objectives

Previous studies have explored O-ISL technology in LEO constellations using different simulation environments and setups, deriving key insights to refine the Concept of Operations (ConOps) and establish system and design requirements. Each simulation tool employed in prior research has its own strengths and limitations, making it particularly suited for specific types of analysis. OptiSystem enables detailed modelling of optical links at a level of precision unmatched by traditional satellite simulation tools; however, it does not incorporate spacecraft dynamics or environmental effects. Conversely, STK and GMAT are powerful tools for simulating constellation dynamics and efficiently generating access reports, yet they lack the capability to model O-ISLs in detail, including laser payload characteristics and attitude control requirements. While both STK and GMAT facilitate similar analyses, a key distinction is that STK is a proprietary software requiring licensing, whereas GMAT is open-source and freely available, albeit with a steeper learning curve. Additionally, despite their ability to model the space environment, these tools simulate satellites at a high level without subsystem granularity, preventing the detailed modelling of satellite operations such as command execution and real-time interactions with a ground segment.

This work introduces an innovative approach to modelling O-ISL in LEO by leveraging a self-developed simulation environment built using the NASA Operational Simulator for Small Satellites (NOS3) [14]. NOS3 provides a high-fidelity simulation framework for CubeSat and small satellite missions, integrating flight software, satellite subsystem models, ground control software and an environmental simulator. This integration allows for a more realistic representation of satellite operations compared to previously used tools.

The objective of this paper is to present the implementation of the O-ISL subsystem model within NOS3 and demonstrate how this environment can be used to evaluate the operational impact of O-ISLs on a LEO constellation. To assess the system's effectiveness in time-sensitive applications, various operational scenarios are simulated, including real-time data exchange between satellites and data relay to ground stations. In particular, a scenario is explored in which a satellite collecting critical information over an area without ground station coverage relays the data through O-ISLs to a satellite with ground access, thereby minimizing data latency. Furthermore, it is highlighted how NOS3's deeper simulation capabilities allow for the exploration of O-ISL aspects not accessible in other simulators, while also discussing its current limitations and potential future improvements.

## 1.1 Structure of the Paper

Section 2 outlines the system description and development process of the operational simulator used in this study. NOS3's general architecture is described, along with the integration of the O-ISL payload into the simulator, including the design of attitude pointing modes and the constellation simulation setup. Section 3 focuses on the operational scenarios tested using the tool, such as data relay to ground stations, the exchange of telecommands (TC) and telemetry (TM) between CubeSats, and the routing of large data volumes within the orbital ring. Section 4 presents key insights gained from these operational scenarios. Section 5 discusses the lessons learned from using NOS3 as an operational simulator and provides an overview of future extensions to the O-ISL simulation environment. The conclusions and a summary of the results are presented in Section 6.

## 2. Operational Simulator and O-ISL Implementation

For this study, NOS3 is used to model an orbital ring of a Walker-Delta LEO constellation composed of 8U CubeSats at 500 km of altitude and 70 degrees orbital inclination. Each CubeSat in the constellation is simulated with emulated models of core spacecraft subsystems, integrated with NASA's core Flight System (cFS) [15] and communication protocols that link the flight software to the subsystem hardware models via their respective interfaces. The environment simulator utilized is NASA's 42 [16], which is directly integrated into NOS3. This provides orbital propagation and attitude information for the satellites and generates environmental data relevant to the spacecraft components. Additionally, NOS3 enables ground segment simulation via OPEN-C3 Cosmos [17], allowing the transmission of TC and reception of TM between the satellites and the ground segment. This feature makes NOS3 a unique and comprehensive simulation tool, capable of modelling and operating spacecraft in a realistic manner. Fig. 1 displays the basic architecture of NOS3.

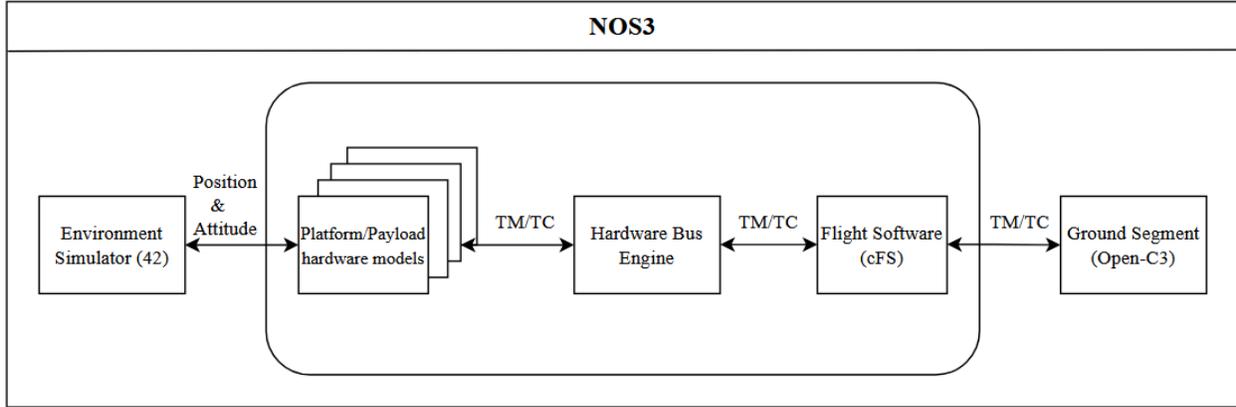


Fig 1: NOS3 General Architecture.

By default, each NOS3 satellite runs on an isolated Virtual Machine (VM) with its own ground segment. Thus, one of the limitations of NOS3 in this version, is that only one satellite can be commanded at a time. While other approaches have been suggested to extend the control capability to multiple spacecrafts [18], this work has been conducted using the default NOS3 configuration. Section 2.3 details how inter-satellite interactions were managed across multiple VMs. To simulate an orbital ring of 24 satellites, multiple NOS3 instances were deployed and a custom O-ISL subsystem was integrated on each satellite within the constellation. NOS3 provides a basic set of platform subsystems for spacecraft modelling. Of particular relevance to this work, the Attitude Determination and Control System (ADCS) includes attitude sensors like fine sun sensors, coarse sun sensors, a star tracker and an inertial measurement unit, as well as actuators such as reaction wheels and thrusters. A GPS receiver is also implemented. However, the O-ISLs subsystem used to conduct this study has been entirely developed in-house and is described in detail in the following section.

### 2.1 Modelling of the O-ISL Subsystem

The modelling of the O-ISL subsystem is based on TESAT’s SCOT-20 optical terminal datasheet [19], which is specifically designed for CubeSats operating in LEO orbit. The SCOT-20 terminal features a fine steering mirror and coarse pointing assembly to handle fine adjustments within a Field of Regard (FoR) of  $\pm 1$  degree. The main features of this terminal are summarized in Table 1.

Table 1: Key characteristics of the SCOT-20 optical terminal modelled in this study.

Parameter	Mass	Size	FoR	Data Rate	Range
Value	1.6 Kg	1 U	$\pm 1$ degree	100 Mbps	Up to 2000 km

Each CubeSat is equipped with two laser communication terminals mounted on the ram and wake directions, enabling it to establish intra-planar links with either the forward or backward adjacent satellite within the same orbital plane. Due to the narrow FoR, a continuous "Ring" O-ISL configuration, where each CubeSat maintains simultaneous links with both its forward and backward neighbours, is not feasible. This limited FoR also imposes strict pointing requirements on the ADCS, which must ensure precise attitude adjustments before establishing the link. Given the study's focus on the operational aspects of O-ISLs, the modelling approach was simplified by neglecting the detailed dynamics of the pointing, acquisition, and tracking system, as well as the mechanical design of the terminals. A further simplification is the use of the Universal Asynchronous Receiver Transmitter (UART) protocol for communication between the laser terminals and the flight software, instead of the Controller Area Network (CAN) bus system.

#### 2.1.1 Attitude Pointing Mode Design

Operating O-ISLs necessitates precise pointing and tracking mechanisms due to the narrow beamwidth of laser communications. Accurate knowledge of satellite position and attitude is critical. In this approach, onboard orbital propagation is used to predict the positions of each satellite’s forward and backward neighbours in the Earth-Centered Inertial (ECI) frame, based on periodically updated Two-Line Elements. Each CubeSat determines its own position

using GPS. To acquire the necessary pointing, each CubeSat must align with its designated target, be it a ground station or another CubeSat. The ADCS operates in two distinct O-ISL modes:

1. **ISL Mode:** The CubeSat points and tracks either its forward or backward neighbour within the orbital plane.
2. **OGS Mode:** The CubeSat aligns one of its laser terminals with an OGS.

The pointing algorithm follows these steps:

1. Compute the target's position in the ECI frame using onboard orbital propagation (for ISL Mode) or retrieve the known OGS location (for OGS Mode).
2. Retrieve the CubeSat's own position via GPS and determine its attitude using the star tracker and sun sensors.
3. Compute the inter-satellite vector in ECI (for ISL Mode) or the line-of-sight vector to the OGS (for OGS Mode), then transform it into the body frame using the star tracker quaternion.
4. Align the CubeSat's payload body axis with the previously calculated unit vector.
5. Maintain alignment using a Proportional-Derivative (PD) control algorithm.

The PD algorithm ensures precise pointing by continuously comparing the desired and actual pointing directions. When the vectors are nearly identical, the pointing error is set to zero. Otherwise, the cross product of the vectors generates an orthogonal error vector, which is used to adjust the CubeSat's orientation. The closed-loop attitude control system limits the attitude slew rate, computes the angular velocity error, and determines the necessary corrective torque via the PD control law. This torque command is then applied through the reaction wheels, ensuring stable and accurate link maintenance. This method achieves a pointing accuracy better than 1 degree, enabling reliable communication between CubeSats in ISL mode. Figure 3 illustrates how the pointing accuracy, measured as the scalar product between the terminal axis and the ISL unit vector in the body frame, remains consistently within the required threshold.

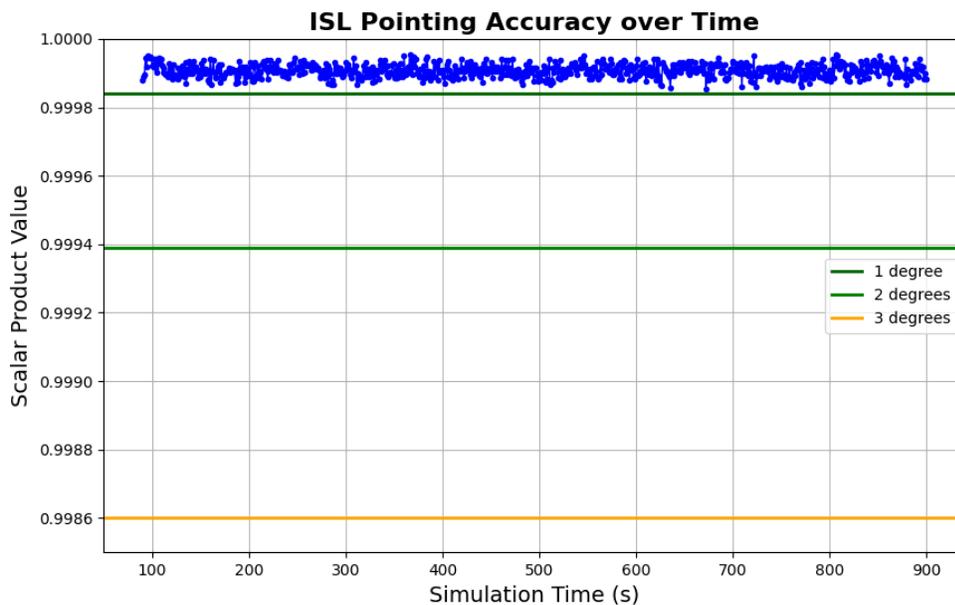


Fig. 3: ISL mode Pointing Accuracy over 900 simulation seconds.

In OGS Mode, pointing accuracy varies significantly, ranging from 1 degree to as much as 11 degrees when the CubeSat passes directly overhead, as shown in Figure 4.

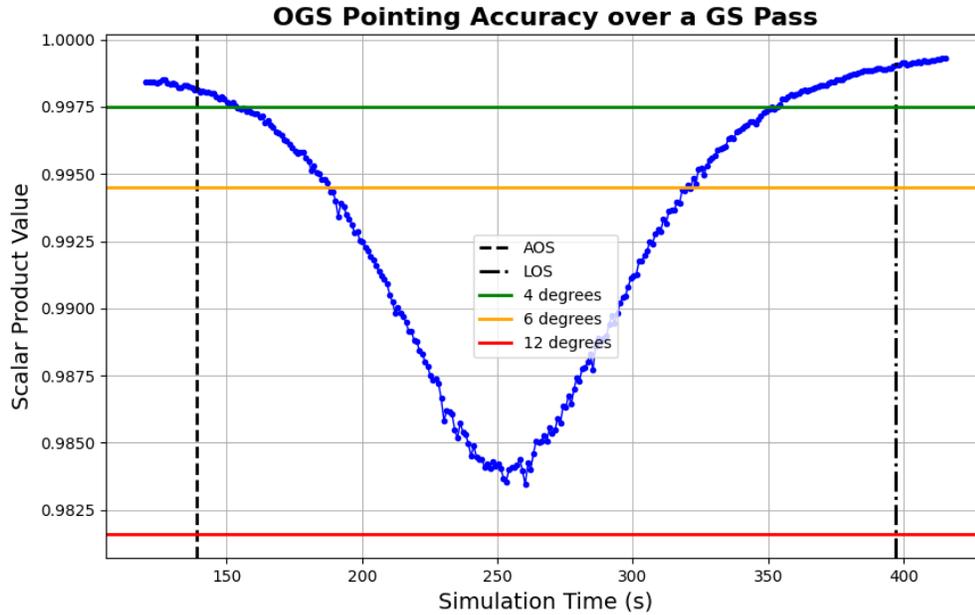


Fig. 4: OGS mode pointing accuracy during a pass above the ground station. Acquisition of Signal (AOS) indicates the start of the pass while Loss of Signal (LOS) indicates its end.

This degradation is primarily due to the high relative velocity between the CubeSat and the ground station, particularly when the satellite is flying perpendicular to the OGS. The challenge is further exacerbated by the fact that in this specific orbital plane, the Earth's rotation is opposite to the CubeSat's direction of revolution, making precise tracking particularly difficult. This issue is not unexpected, as ADCS control strategies for maintaining stable optical links with ground stations remain an active area of research. Given that this study focuses on modelling O-ISLs for operational scenarios, the OGS pointing accuracy requirement has been relaxed to 12 degrees. This adjustment ensures a continuous optical downlink throughout the entire pass.

### 2.1.2 TM/TC and Data Transfer Protocol

NOS3 allows the creation of custom TC and TM parameters for each subsystem. For the O-ISL subsystem, a key TC is the "Require Data" command. This command is automatically issued by the flight software's scheduler to the terminal's hardware models at a fixed interval of one second, ensuring periodic updates of TM parameters. These parameters include the inter-satellite vectors for the forward and backward neighbours in the ECI frame, Boolean flags indicating successful pointing and tracking, and the status of an active connection. Additionally, available onboard storage is monitored and updated, particularly during large file transfers, with a maximum payload data capacity set to 8 GB. The most fundamental command used in this study is the "Data Transfer" command, used to exchange any type of data, including TM/TC and large files, from one CubeSat to the other. The protocol used to exchange data is the CCSDS File Delivery Protocol (CFDP) [20], a standardized protocol designed for reliable data transfer in space missions. CFDP segments data into discrete files, which are transmitted over potentially unreliable communication channels using a store-and-forward mechanism. It employs packet acknowledgments to ensure data integrity and retransmits lost packets when necessary. This capability makes CFDP particularly well-suited for O-ISL communications, where intermittent connectivity and constrained bandwidth are common. Upon receiving a "Data Transfer" command, the CubeSat identifies the designated recipient—either one of its two adjacent neighbours or a ground station—and adjusts its ADCS mode accordingly. Once aligned, it attempts to establish a handshake with the receiver to verify both correct alignment and memory availability. If the receiver is already tracking the sender, the handshake is immediately successful, and communication begins. Otherwise, a beacon signal prompts the receiver to align. Following successful alignment, data transfer is simulated by generating packets with headers and trailers, along with simulated acknowledgment signals to confirm successful transmission. The tracking of the receiver is actively maintained throughout the transfer process. Fig. 5 illustrates the logic of the O-ISL file transfer between two satellites after the receipt of the "Data Transfer" command.

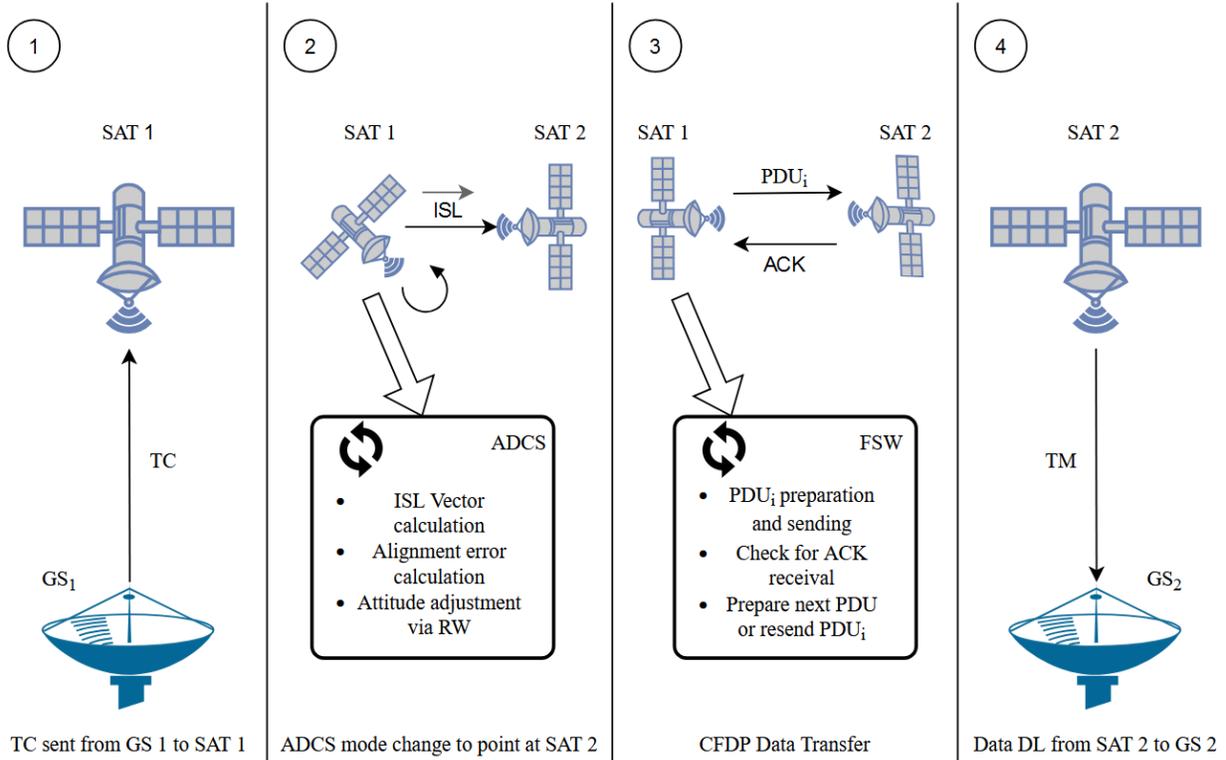


Fig. 5: Operational example of CFDP usage for the transfer of time-sensitive data via O-ISL.

This command has been used in this study to simulate the different operational scenarios explained in Section 3.

## 2.2 Constellation Simulation Setup using VMs

In the current NOS3 framework, simultaneous commanding of multiple satellites and direct inter-satellite interactions within a single simulation are not supported. Each CubeSat operates within an independent VM, along with its own ground segment simulation. Consequently, inter-satellite communication is achieved externally through server-level shell scripts. A primary script facilitates data exchange between nearby satellites in the orbital ring by managing the transfer of alignment and memory utilization data. Each satellite generates a status file every second, containing information about its orientation relative to adjacent CubeSats and its onboard memory usage for payload data. This file is then transmitted to neighboring satellites, enabling mutual data sharing. Using this mechanism, the simulation determines when two CubeSats are correctly aligned, thereby triggering the O-ISL transfer procedure. A secondary script governs the visualization of satellite interactions during a transfer event. When a CubeSat engages in an O-ISL, either as a sender or receiver, a Boolean TM flag is set to true. This action creates a file inside the 42 simulation folder within the VM, activating a script responsible for modifying the camera view. Within 42, the laser communication link is visually represented by a red beam extending from the sender's active laser terminal to the receiver's terminal. This visualization is applied symmetrically to both transmitting and receiving satellites. Figure 6 illustrates the laser beam representation in NOS3 from both perspectives.

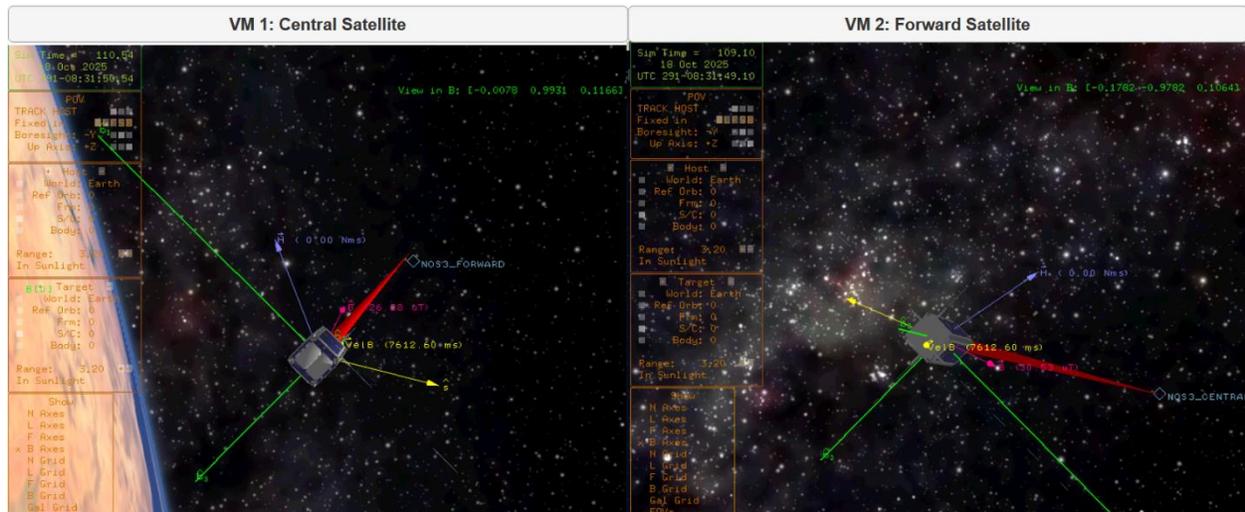


Fig. 6: O-ISL visualisation in the 42 Cam.

In addition to these scripts, another script must be executed within each VM to evaluate the information received via O-ISL and take appropriate action based on a specific ID flag embedded in the data. The receiving satellite must be capable of handling data according to its type. In the case of TC, the command must be forwarded to the FSW for execution. Conversely, for TM or OGS data relay, the information must be correctly processed or forwarded to the next satellite in the orbital ring for further transmission.

### 3. Simulated Operational Scenarios

This section outlines the different operational scenarios simulated using the architecture described in the previous section. These scenarios involve the exchange of TM and TC between CubeSats and data relay to ground stations, including the transfer of large amounts of data that requires its division and routing across multiple nodes of the orbital ring.

#### 3.1 TM/TC Sharing via O-ISL

In this scenario, CubeSats exchange TM or TC within the orbital ring via O-ISL. TM data can be shared between the CubeSats, and eventually with an OGS, for different purposes:

- **Health and Status Monitoring:** To monitor subsystem states such as battery health, onboard computer CPU load and memory usage, among others.
- **Sensor Data:** For time-sensitive payload data, such as imaging or spoofing detection, that needs to be rapidly sent to a ground station or another satellite for analysis and response.
- **PNT Data:** PNT data, such as attitude and location information, can be used for Precise Orbit Determination (POD) and time synchronisation via O-ISLs.

The TC to be shared could be flight control commands, payload or subsystem's specific commands, communication and link management commands and so on. The simulation environment adopts a specific Extensible Markup Language (XML) format to discriminate between the different data types. The XML header includes the sender satellite's identifier, a timestamp in UTC, and source and destination folder paths for file transfer. The Content field contains the actual data. This XML format is shared by both the server scripts that move files between satellites' repositories and the internal VM scripts that process received files, ensuring efficient communication. To simplify the approach, any error detection method like checksums and cyclic redundancy checks are not included in this logic.

An interesting aspect to analyse is the amount of time it takes the ADCS subsystem to switch between pointing modes. A CubeSat in this scenario must quickly change its pointing to ensure rapid data processing. Table 2 presents the average time required for a CubeSat to switch between ADCS modes in the most common operational scenarios.

Table 2: Average transition time between different ADCS mode for common operational scenarios. The Nadir pointing mode refers to a configuration where the optical payload is oriented perpendicular to Earth’s surface with an accuracy of 1 degree. The transition from OGS pointing to ISL mode occurs either at the start or at the end of a pass.

Initial ADCS mode	Final ADCS mode	Average Transition Time
Nadir pointing ( $\pm 1$ deg)	Either ISL mode ( $\pm 1$ deg)	110 seconds
OGS mode ( $\pm 12$ deg)	Either ISL mode ( $\pm 1$ deg)	87 seconds
ISL mode Forward ( $\pm 1$ deg)	ISL mode Backward ( $\pm 1$ deg)	55 seconds
Sun safe pointing ( $\pm 1$ deg)	Either ISL mode ( $\pm 1$ deg)	43 seconds

### 3.2 O-ISL Data Relay

This scenario simulates a satellite acquiring actionable information over an area of interest but lacking direct ground station coverage. To minimize data latency, the satellite relays the data via O-ISLs within the constellation to another satellite with ground station access. In the simplest case, the data volume is small enough to be entirely downlinked by a single satellite during a pass. For the modeled constellation at 500 km altitude, assuming a minimum elevation angle of 30 degrees for optical downlink, the average pass duration ranges between 100 and 250 seconds, depending on the ground track at the specified simulation time. Assuming the maximum pass duration of 250 seconds and a downlink rate of 100 Mbps, the theoretical maximum data volume that can be transmitted during a pass is 3.125 GB. However, to account for factors such as link acquisition delays, pointing inaccuracies, atmospheric effects, and unexpected interruptions, a configurable safety margin was introduced. This margin, set to 20% of the pass duration, corresponds to a 50-second buffer for a 250-second pass, reducing the effective downlink time to 200 seconds. As a result, the maximum data volume that can be reliably downlinked per pass, henceforth referred to as the "downlink capacity", is approximately 2.5 GB. If the total data volume exceeds this 2.5 GB limit, the excess data must be split into smaller segments and dynamically routed through the constellation to other satellites with available downlink opportunities. This ensures the lowest possible latency while maintaining a reliable data transfer process. The next subsection will explore the logic of the routing algorithm implemented in this simulation.

#### 3.2.1 Routing Algorithm

This work implements a decentralized decision-making routing logic, where each CubeSat autonomously determines how to route information upon receipt, with the main goal of reducing the latency. To avoid inefficient exchanges, where a receiver might return data to the original sender due to the unavailability of the next node in the orbital ring (e.g., after passing over the desired OGS), a safety routing margin is introduced. This safety margin accounts for potential connection losses during the O-ISL and the time required for CubeSats to switch between ADCS modes (see Table 2). The decentralized nature of the system allows the nodes to dynamically adapt and reassess the routing path, enhancing system robustness by avoiding single point of failure. Upon receiving data, each CubeSat evaluates the size of the data and the visibility predictions for the desired OGS. If a CubeSat is about to pass over the OGS and the data size is manageable within the downlink capacity, it switches to OGS tracking mode for downlink. If the data size exceeds the CubeSat's capacity or if another CubeSat in the constellation has an earlier visibility window, the CubeSat assesses the pass predictions of other satellites in the constellation and assigns downlink sizes based on pass duration, selecting the most suitable node for data transfer. If the data size exceeds a node's downlink capacity, it is divided into smaller chunks, with each node assigned a portion it can handle.

A second validation step ensures that the total routing time does not exceed available resources, potentially requiring the addition of another node to the path. The routing process iterates through each node, with data transferred to the OGS in a cumulative manner. Fig. 7 illustrates an example of the OGS relay for transferring 8 GB of data, starting from the central Node 0. Assuming each satellite has enough available memory, the data portion assigned to each spacecraft is based on its downlink capacity. Node 0 will first route the data forward to the next node, ensuring that the CubeSat with the earliest pass over the desired OGS has enough time to receive the data through the hops and initiate the downlink. Once this is completed, it will route the data backward to the previous node before switching to OGS tracking mode to await the start of its own pass.

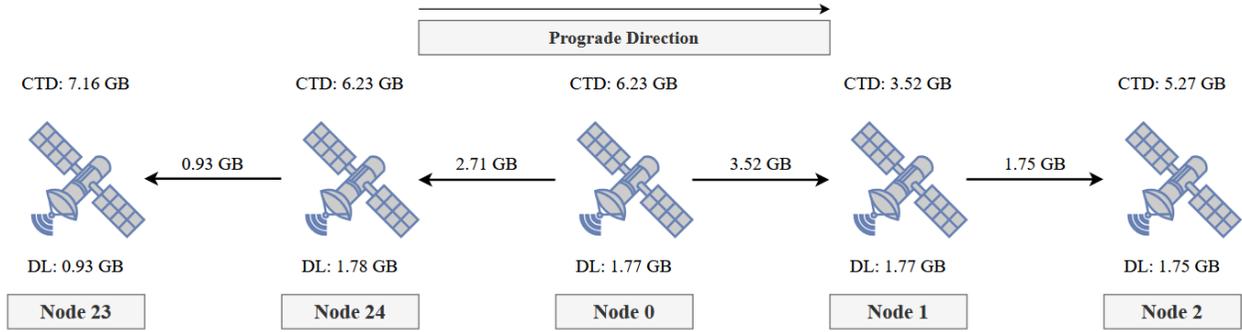


Fig. 7: 8 GB Data division across the satellites in the orbital ring. The downlink size (DL) coincides with the downlink capacity and the Cumulative Transferred Data (CTD) indicates the total amount of data routed through the ring before reaching a specific node and initiate the downlink preparation.

If a node's memory is full, the sender will transmit only the portion of data it can store, resuming transfer once additional memory becomes available. This iterative process ensures that each node makes autonomous decisions, and the data is cumulatively downlinked to the desired OGS. Fig. 8 summarizes the overall logic of the routing process.

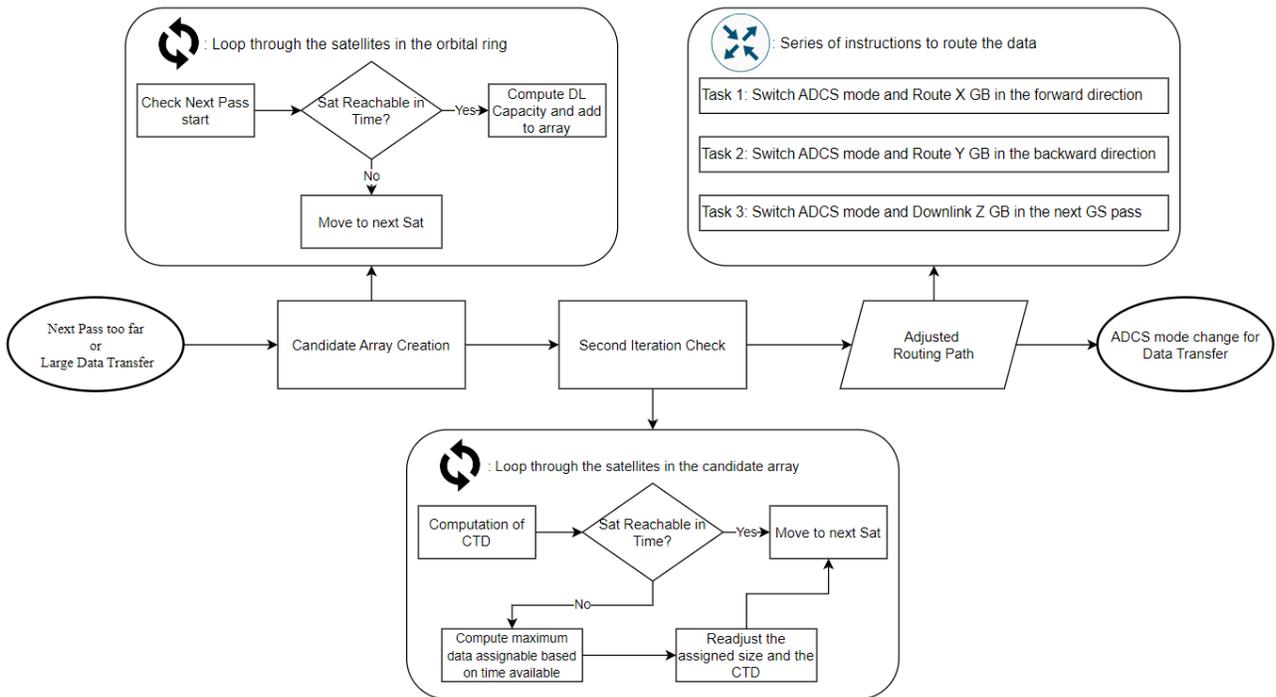


Fig. 8: Flowchart of the Routing Algorithm.

#### 4. Key Insights from the Simulated Operational Scenarios

The results from these simulations highlight several critical factors influencing the performance and efficiency of data transfer within the CubeSat constellation. These insights are summarized as follows:

- **Downlink Capacity as a Latency Bottleneck:** The primary factor influencing latency is the downlink capacity, which is intrinsically tied to the constellation design. When transferring large files, it is inefficient to assign data portions exceeding a satellite's downlink capacity, typically below 2 GB for a LEO constellation, since this would prevent successful transmission within a single pass, requiring additional routing in the backward direction, potentially returning data to the original sender. This acts as a key bottleneck for the routing process, as larger files necessitate the involvement of multiple satellites. Constellation design plays a crucial role in mitigating this limitation: higher-altitude constellations allow for longer pass durations and larger downlink

capacities, reducing the number of satellites required. However, increasing altitude also results in longer signal round-trip times, requiring a trade-off between latency and efficiency based on mission-specific needs. A potential solution to this bottleneck is the use of laser communication terminals with adaptive data rates, which can vary depending on whether the link is unidirectional or bidirectional. While this study models a terminal with a fixed data rate, many existing systems offer higher data rates in unidirectional mode, particularly for downlinking data to an OGS. In this case, the primary limitation shifts from the satellite's downlink capacity to the amount of data that can be routed to the satellite with the earliest ground pass, which ultimately depends on the constellation density.

- **Need for Autonomous Inter-Satellite Alignment in Large Data Transfers:** To minimize latency, large data volumes must be relayed across multiple hops within the orbital ring. However, the larger the data size, the longer the transfer time, necessitating data division into forward and backward portions. Ensuring synchronization between these portions requires an autonomous alignment mechanism, such as a beacon signal, to dynamically coordinate the relative positioning of forward and backward satellites. This reduces reliance on operator supervision and shifts network management responsibilities from the ground segment to the space segment.
- **Challenges in Decentralized Decision-Making:** Unlike large-scale satellite constellations like Starlink, CubeSats operating with O-ISLs face unique challenges due to their limited ability to maintain multiple simultaneous links. Since a CubeSat can typically establish an O-ISL with only one satellite at a time, decentralized decision-making becomes even more complex. Without a structured oversight mechanism, there is a higher risk of inefficient routing, network congestion, and communication delays. A CubeSat must carefully decide when and with whom to establish an O-ISL, as breaking a link to form a new one could lead to temporary loss of connectivity. To mitigate these challenges, an external network management entity is required to monitor data flow across all CubeSats, optimise link scheduling and ensure load balancing. While strategic oversight from the ground remains necessary, simulation environments like the one described in this study are essential for developing routing strategies and optimising network performance. These tools help transition CubeSat networks from being heavily reliant on ground-based coordination toward a more autonomous, in-space decision-making system.
- **Impact of Safety Margins on Routing Efficiency:** As described in Section 3.2.1, the routing algorithm incorporates a safety margin to account for potential connection losses during O-ISLs and the time required for CubeSats to transition between ADCS modes. While a larger safety margin enhances reliability, excessive conservatism results in inefficiencies by increasing overall delivery time, i.e. the total time required to downlink all data from the originating satellite to the nearest available OGS. Multiple tests were conducted to evaluate the impact of varying safety margins on routing performance and total delivery time. Figure 9 presents the results, showing delivery time as a function of different safety margins and file sizes.

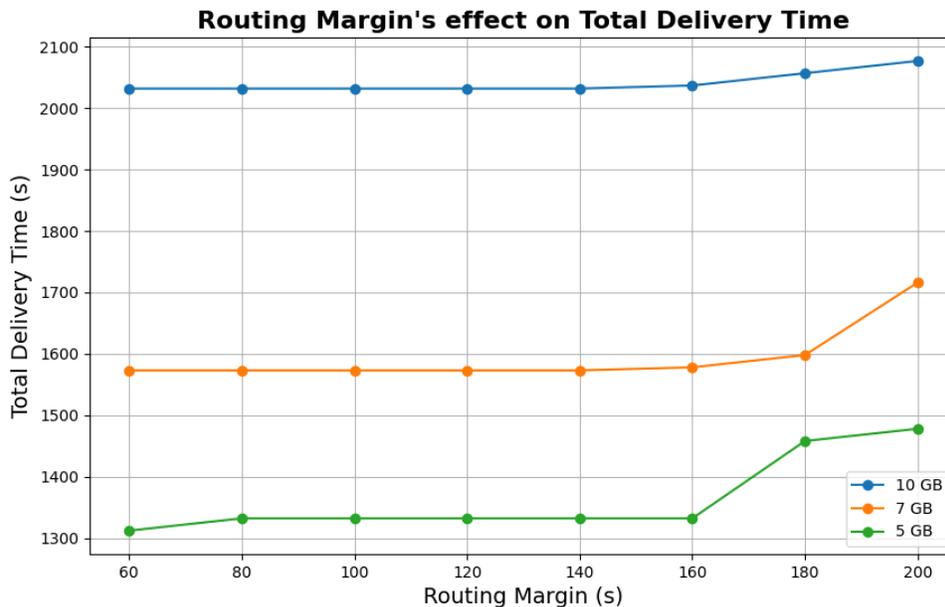


Fig. 9 Total delivery time as a function of the routing margin for three different data sizes.

In this specific study, the central node, located over Kalabo, Zambia (Latitude: -14.9701, Longitude: 22.6814), was tasked with downlinking large data volumes to an OGS in Igrim, Russia (Latitude: 63.19, Longitude: 64.42). The total delivery time represents the elapsed time from data acquisition to the completion of the final downlink. The figure demonstrates that total delivery time remains largely unaffected by safety margins within the range of 60 seconds (80 seconds for 5 GB) to 160 seconds. However, excessively conservative margins increase delivery time significantly, as the satellite with the earliest upcoming pass over the target OGS may be deemed unreachable, forcing data redistribution to further nodes. Conversely, reducing the margin below 140 seconds introduces additional risks without offering reductions in delivery time. Based on these findings, the simulation environment adopts a default safety margin of 120 seconds, providing an optimal balance between reliability and efficiency.

Furthermore, in the same scenario, 8 GB of payload data acquired over Kalabo could be fully downlinked to a single ground station in Igrim within approximately 29 minutes by utilizing O-ISLs across a network of five CubeSats. Achieving the same delivery time without O-ISLs would require at least four additional ground stations along the CubeSat's ground track, significantly increasing both mission costs and complexity. This further underscores the role of O-ISLs in reducing latency and enhancing the efficiency of time-sensitive operations. Moreover, this simulation framework allows for the evaluation of how different laser terminals impact overall data delivery time. For instance, adopting a more powerful terminal with a FoR of  $\pm 60$  degrees in elevation and  $\pm 175$  degrees in azimuth and a data rate of 1 Gbps, would enable a continuous ring geometry, eliminating the need to switch between ADCS modes for inter-satellite links. As a result, routing safety margins could be removed, ensuring a constant O-ISL network and further reducing data delivery time to just 6 minutes and 18 seconds. In addition, having a full ring geometry would allow to easily control the CPU load and health condition of each CubeSat in the constellation in real-time, simplifying significantly the network management. However, this improvement comes at the cost of increased mass and power requirements for each CubeSat.

## 5. Lessons Learned Using NOS3 as an Operational Simulator

The modelling of the O-ISLs and the testing of operational scenarios have been conducted using NOS3, which, as detailed in Section 2, offers unique capabilities for interacting with spacecraft through TM/TC and for modeling spacecraft components and attitude modes in great detail. However, several limitations and challenges have been identified in using NOS3 for constellation-level simulations, which merit discussion.

One significant limitation is that NOS3 currently supports the command of only a single satellite at a time. This restriction presents a challenge for simulating constellations, as it complicates the modelling of inter-satellite interactions and prevents the real-time synchronization needed for operational satellite networks. As a result, simulating coordinated constellation behaviours, such as dynamic routing of TM/TC across multiple satellites, becomes challenging. Moreover, NOS3 operates within VMs, which can be cumbersome due to their independent nature and difficulty in scaling for larger systems. These VMs place additional constraints on performance, particularly in terms of hardware resource consumption. When running simulations with a single, detailed satellite model, the computational requirements are substantial, limiting the maximum simulation speed and the number of satellites that can be simulated concurrently. In this work's tests, even on a high-performance server (2 x 32 cores, 512 GB RAM), it is not possible to achieve real-time simulation speed, with a maximum of approximately 16 satellites simulated at once. Furthermore, the learning curve associated with NOS3 can be steep, particularly for aerospace engineers without extensive expertise in embedded software engineering or real-time simulations. This adds an additional layer of complexity when adopting the tool for operational analysis.

Despite these challenges, it is important to note that the NASA NOS3 team is actively addressing these limitations. One improvement is the completed transitioning from VMs to Docker containers, which enhances scalability and computational efficiency. Additionally, the team aims to support the command and control of multiple satellites simultaneously, enabling more realistic constellation simulations. In terms of usability, the team is working to expand documentation to facilitate easier adoption and integration. Notably, NOS3 has already transitioned to YAMCS (Yet Another Mission Control System) [21] for ground software, which offers built-in support for industry standards such as CCSDS, CFDP, and XTCE (Telemetric and Command Exchange). With these planned updates, the next stages of our analysis, as outlined in the following section, will be conducted using an enhanced and more robust version of NOS3, allowing for more comprehensive and realistic simulations.

### 5.1 Future Extensions and Improvements

Our goal is to continue expanding this O-ISL simulation environment within NOS3 to support a broader range of operational scenarios applicable to various mission profiles. Additionally, we aim to incorporate details that were initially simplified for the purposes of this analysis. The following areas represent key focus points for future developments:

- **Modelling of Cloud Coverage.** In the current simulation, we assume the constant availability of all OGSs. Moving forward, we plan to integrate cloud coverage modelling, utilizing weather forecast databases previously employed in other DLR projects. This will allow us to simulate more realistic operating conditions, accounting for potential interruptions caused by atmospheric obstructions.
- **Enhanced ADCS Mode for OGS Downlink.** As detailed in Section 2.1.1, the current ADCS mode used for OGS downlink when the OGS is moving opposite to the satellite's orbit exhibits suboptimal performance. One of our priorities for future work is to significantly improve the pointing accuracy in such scenarios to ensure more reliable communication links.
- **Link Budget Computations.** Our current model of the optical system does not take link budget computations into account. However, these calculations are crucial for understanding the feasibility and performance of O-ISLs in operational contexts. Incorporating link budget models is a high priority for future iterations to ensure that the simulation accurately reflects real-world constraints and performance limitations.
- **Implementation of Inter-Planar O-ISLs.** The present study focuses solely on intra-planar O-ISLs, which are stable and relatively easy to establish and maintain. However, in practical applications, inter-planar O-ISLs, while more challenging to implement, could offer significant advantages to the overall ConOps. Future work will explore the feasibility of inter-planar links, including strategies for handovers between satellites and methodologies for establishing and maintaining such links.
- **Introducing methods for PNT applications.** NOS3 provides capabilities to simulate operational scenarios for PNT applications involving O-ISLs, such as POD and Orbit Determination and Time Synchronisation (ODTS). We plan to extend the simulation environment to allow for the execution of PNT commands, enabling us to investigate the practical implementation and performance of these applications within our setup.

## 6. Conclusions

This paper presents the first operational simulator for O-ISLs in LEO constellations built using NASA's NOS3 open-source. Following an overview of commonly used constellation and optics simulation tools, the unique benefits and opportunities offered by NOS3 are highlighted. These include the ability to model the spacecraft holistically, incorporating hardware models, environment simulation, flight software, and ground segment. This enables operational interaction with the spacecraft, allowing TC to be sent and TM to be received.

The O-ISLs have been modeled as 1U optical payloads with a 1-degree field of view, mounted on the CubeSat's ram and wake directions to establish intra-planar links with forward and backward neighbors within the same orbital plane. Various ADCS modes have been implemented to enable pointing and tracking of either a nearby satellite or a ground station. Additionally, multiple TM parameters and commands have been incorporated to simulate operational scenarios, including TC/TM exchange and data relay to ground stations. A simplified CFDP protocol has been implemented to model data exchange between satellites, along with a routing algorithm designed to optimize the division and downlink of large amounts of payload data.

The key insights and requirements derived from the testing of these operational scenarios are presented, along with lessons learned from using NOS3 as an operational simulator. Results demonstrate that a LEO CubeSat constellation can reliably establish and maintain intra-planar links with pointing accuracy better than 1 degree, while challenges remain in tracking a ground station for data downlink. The benefits of O-ISLs in reducing latency and ground segment dependency for downlinking large volumes of data are also discussed.

Future improvements to this simulation environment will expand the range of scenarios to include inter-planar links, cloud blockage modeling, and the integration of precise orbit determination and onboard orbit determination and time synchronization for PNT applications.

## Acknowledgements

The author extends sincere gratitude to colleagues in the Department of Space & Ground Segment Technology within DLR's Galileo Competence Center. Special thanks go to Spencer Ziegler for his invaluable suggestions and perspectives, which proved critical to the success of this work.

## References

- [1] Li, Xingxing & Zihao, Jiang & Ma, Fujian & Lv, Hongbo & Yuan, Yongqiang & Li, Xin. (2019). LEO Precise Orbit Determination with Inter-satellite Links. *Remote Sensing*. 11. 2117.
- [2] Michalak, Grzegorz, Giorgi, Gabriele, Deprez, Cécile, "Enhanced Orbit Determination of GNSSs with Optical Inter-satellite Links," *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, St. Louis, Missouri, September 2021, pp. 2838-2847. <https://doi.org/10.33012/2021.18012>.
- [3] G. Giorgi, B. Kroese, and G. Michalak, "Future GNSS constellations with optical inter-satellite links. Preliminary space segment analyses," in 2019 IEEE Aerospace Conference, March 2019a, pp. 1–13. [Online]. Available: <https://doi.org/10.1109/AERO.2019.8742105>.
- [4] Kur, T., Liwosz, T. & Kalarus, M. The application of inter-satellite links connectivity schemes in various satellite navigation systems for orbit and clock corrections determination: simulation study. *Acta Geod Geophys* **56**, 1–28 (2021). <https://doi.org/10.1007/s40328-020-00322-4>.
- [5] H. Kaushal and G. Kaddoum, "Optical Communication in Space: Challenges and Mitigation Techniques," in *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 57-96, Firstquarter 2017, doi: 10.1109/COMST.2016.2603518.
- [6] Zachary Warren and Renny Fields. 2022. Optical crosslinks and satellite synchronization for GNSS, communications, and beyond. *GPS Solut.* 26, 3 (Jul 2022). <https://doi.org/10.1007/s10291-022-01233-3>.
- [7] Saiyyed, R., Sindhwani, M., Sachdeva, S. *et al.* Enabling high-speed and large-capacity data transmission in optical inter-satellite communication links under various conditions. *J Opt* (2024). <https://doi.org/10.1007/s12596-024-02227-9>.
- [8] OptiSystem, Optiwave, 2025. [Online]. Available at: <https://optiwave.com/optisystem-overview/> [Accessed on 05/02/2025.]
- [9] STK. Systems tool kit (STK), 2025. [Online]. Available at: <https://www.agi.com/products/stk> [Accessed on 04/02/2025.]
- [10] S. Raffa, L. Pizzuto, J. Poliak, "End-to-End Availability Analysis for an Optical-RF Large Constellation Concept", 75th International Astronautical Congress (IAC), Milan, Italy, 14-18 October 2024, IAC-24-B2.4.8.
- [11] A. Baltaci and K. Shortt, "Investigation of the Influence of LEO Constellation Dynamics on Optical Inter-satellite Links," *2023 IEEE International Conference on Space Optical Systems and Applications (ICSOS)*, Vancouver, BC, Canada, 2023, pp. 121-127, doi: 10.1109/ICSOS59710.2023.10490286.
- [12] S. Hughes, D. Conway, J. Parker, Using the General Mission Analysis Tool (GMAT), (2017) 10.13140/RG.2.2.12685.54249.
- [13] Kling, Ulrich Rolf und Pavicic, Sabrina und Hiebl, David und Ziegler, Spencer Wilson (2023) Reducing Data Latency with Inter-Satellite Links in LEO Constellations: Trade-off Analysis and Impact on Concept of Operations. 17th International Conference on Space Operations, 2023-03-06 - 2023-03-10, Dubai, United Arab Emirates.
- [14] NOS3. NASA Operational Simulation for Small Satellites. [Online]. Available at: <https://www.nasa.gov/nasa-operational-simulation-for-small-satellites/> [Accessed on 05/02/2025].
- [15] National Aeronautics and Space Administration, Core Flight System, <https://cfs.gsfc.nasa.gov/>, (accessed 06.02.2025).
- [16] 42, "NASA 42", <https://software.nasa.gov/software/GSC-16720-1>, (accessed 06.02.2025).
- [17] Open C3, "OpenC3 COSMOS", <https://docs.openc3.com/docs>, (accessed 06.02.2025).
- [18] M. Campanelli, "Architecture of a Simulation Test Bench for Operating Large Satellite Constellations", 75th International Astronautical Congress (IAC), Milan, Italy, 14-18 October 2024, IAC-24-B6.5.3.
- [19] TESAT, SCOT20, [https://www.tesat.de/images/tesat/products/240306\\_DataSheet\\_SCOT20\\_A4\\_Druck.pdf](https://www.tesat.de/images/tesat/products/240306_DataSheet_SCOT20_A4_Druck.pdf) (accessed 06.02.2025).
- [20] CCSDS. (2020). *File Delivery Protocol (CFDP) Blue Book* (CCSDS 727.0-B-5). Consultative Committee for Space Data Systems (CCSDS).
- [21] YAMCS, "YAMCS Mission Control", <https://yamcs.org/>, (accessed 17/02/2025)