

Extending the SPECULA Framework for End-to-End Modelling of Bi-Directional Optical Feeder Links

B. Stadler^{a,b}, G. Agapito^c, D. Jenkins^{d,e}, D. Jodlbauer^{a,b}, A. Puglisi^c, F. Rossi^c, R. Biasi^f,
and R. Ramlau^{a,g}

^aRICAM - Austrian Academy of Sciences, Altenberger Strasse 69, 4040 Linz, Austria.

^bMathConsult, Altenberger Strasse 69, 4040 Linz, Austria.

^cINAF - Osservatorio Astrofisico di Arcetri, Largo E. Fermi 5, 50125 Firenze, Italy.

^dCentre for Advanced Instrumentation, Department of Physics, Durham University, UK.

^eEuropean Space Agency-ESA/ESTEC, CSC-SQ, Keplerlaan 1, NL-2201 AZ Noordwijk,
The Netherlands.

^fMicrogate, Via Waltraud-Gebert-Deeg 3e, 39100 Bolzano, Italy.

^gJohannes Kepler University, Altenberger Strasse 69, 4040 Linz, Austria.

ABSTRACT

Free-Space Optical Communication links are essential for meeting the multi-gigabit throughput demands of next-generation satellite networks. However, optical wave propagation through Earth's atmospheric turbulence induces severe phase distortions, scintillation, and spatial coherence loss, limiting fiber coupling efficiency and uplink power delivery. While Adaptive Optics can mitigate these degradation effects, bi-directional feeder links introduce unique challenges such as point-ahead anisoplanatism, dynamic telescope slewing, and transient propagation delays. This paper presents the extension and adaption of SPECULA, an open-source, object-oriented Python framework, for end-to-end bi-directional link modeling. SPECULA provides high-fidelity simulation of wave propagation through multi-layer atmospheric turbulence for both Geostationary and Low Earth Orbit scenarios, accounting for Point-Ahead Angles, effective wind vectors, and path delays. The framework supports closed-loop post-compensation at the receiver and pre-compensation at the transmitter using various reference sources (downlink lasers, Laser Guide Stars, Natural Guide Stars) and wavefront sensor architectures. Designed with a modular, YAML driven structure, SPECULA leverages CuPy and CUDA streams for GPU acceleration and multi-node scaling via MPI. Representative communication link scenarios demonstrate the framework's capabilities, yielding performance metrics such as the Strehl ratio and uplink laser power loss to support the design and optimization of future optical feeder link architectures.

Keywords: Free Space Optical Communication, Bi-directional Feeder Links, Adaptive Optics, End-to-end Simulations, Physical Modelling, Numerical Simulations

1. INTRODUCTION

With the exponential growth in global demand for ultra-high throughput satellite networks, traditional radio frequency feeder links are increasingly unable to meet future throughput demands.¹ Free-Space Optical Communication (FSOC) has emerged as the primary candidate for next-generation satellite feeder links, offering multi-gigabit to terabit data rates, high spatial selectivity, immunity to electromagnetic interference, and access to unallocated spectrum.²⁻⁵ However, optical waves traversing the Earth's atmosphere suffer from spatiotemporal refractive-index fluctuations induced by atmospheric turbulence. This manifests as optical phase distortion, beam wander, scintillation, and spatial coherence degradation, severely reducing single-mode fiber coupling efficiency at the receiver and restricting power delivery on the uplink.

To mitigate these turbulence induced degradations, Adaptive Optics (AO) systems, originally developed for astronomical imaging, have been adapted for ground-to-satellite optical links. While astronomical AO

Further author information: (Send correspondence to B. Stadler)

B. Stadler: E-mail: bernadett.stadler@ricam.oeaw.ac.at

primarily addresses one-way downlink phase correction, bi-directional optical feeder links introduce a significantly more complex operational scenario. In a two-way system, the ground terminal must simultaneously receive a downlink signal and transmit an uplink beam to a moving space asset in Geostationary (GEO) or Low Earth Orbit (LEO). By leveraging the reciprocity of optical wave propagation through atmospheric turbulence,^{6,7} the ground transmitter can pre-compensate the uplink beam using phase measurements derived from the incoming downlink wavefront. However, owing to the finite speed of light and relative satellite velocity, the uplink beam must be directed ahead of the apparent downlink position by a Point-Ahead Angle (PAA).⁸ This spatial displacement causes anisoplanatism,⁹ i.e., the downlink and uplink beams traverse distinct atmospheric turbulence volumes.

Many existing numerical wave optics frameworks and AO simulation tools remain predominantly tailored to unidirectional propagation or astronomical observation scenarios.^{10–13} Consequently, they often simplify or omit telecommunication specific phenomena, such as non-reciprocal path geometry, dynamic telescope slewing, transient propagation delays, and the tightly coupled interplay between downlink wavefront sensing and uplink pre-distortion. Alternative tools prioritizing accessibility and rapid prototyping¹⁴ frequently lack the computational optimization required for heavy, large-scale Monte Carlo simulations. While semi-analytical alternatives, such as the Fourier-domain Adaptive Optics Simulation Tool (FAST),¹⁵ offer substantial computational efficiency, they rely on idealized statistical assumptions. Such models struggle to resolve granular spatial metrics, including instantaneous Point Spread Functions (PSFs). Moreover, they lack the architectural flexibility to incorporate non-standard geometries or novel control loops without exhaustive mathematical re-derivation. Consequently, accurate performance evaluation under severe atmospheric scintillation necessitates robust Monte Carlo end-to-end wave optics frameworks.

To bridge this gap, this paper presents the adaptation and expansion of SPECULA¹⁶ (Scalable Parallel Execution of Computations Upscaling Large Adaptive optics simulations) for the design and optimization of bi-directional optical feeder link architectures. Originally developed at INAF-Osservatorio Astrofisico di Arcetri for large-scale astronomical AO systems, SPECULA has been upgraded into a robust, object-oriented Python framework for end-to-end bi-directional FSOC link modeling. By co-simulating downlink wavefront sensing and uplink pre-compensation within a unified temporal loop, SPECULA provides accurate predictions of essential link metrics including Strehl ratio and the power loss of the uplink communication laser across varying turbulence regimes and elevation angles. Key features and simulation capabilities of SPECULA include:

- **High-fidelity physical modeling:** Full integration of physical wave propagation, PAA dynamics, variable telescope slewing rates, and temporal delays between uplink and downlink trajectories. Dedicated utility functions are available to calculate the PAA, effective wind velocity, and optical path delays.
- **Flexible AO loop architectures:** End-to-end closed-loop modeling of both post-compensation at the ground receiver and uplink pre-compensation at the transmitter. The framework supports multiple reference sources, including downlink signals and Laser Guide Stars (LGS), alongside diverse Wavefront Sensor (WFS) architectures, such as Shack-Hartmann, Pyramid, and curvature sensors.
- **Modular system design via YAML files:** Designed for high modularity, SPECULA allows users to fully configure and customize each block of the end-to-end simulation chain via YAML files. This decoupled design enables rapid experimentation, easy component swapping, and seamless reproduction of complex link topologies without modifying core code base logic.
- **Parallelization & Scalability:** Built for high-throughput computations, SPECULA leverages CuPy and CUDA streams for GPU acceleration, scaling seamlessly from single-node workstations to High-Performance Computing (HPC) clusters via Message Passing Interface (MPI) support.
- **RTC Integration:** Bindings to RTC systems based on Artic¹⁷ and DAO¹⁸ have been developed, allowing to include, use, and test the same RTC implementations that run on the real system. Note that these bindings are not yet publicly available.

The remainder of this paper is organized as follows. Section 2 details the theoretical foundation and governing physics of bi-directional atmospheric propagation. Section 3 describes the architecture and modular structure of the SPECULA framework. Section 4 presents validation benchmarks and numerical results and Section 5 concludes the paper.

2. HIGH-FIDELITY PHYSICAL MODELING

2.1 Atmospheric Turbulence and Propagation

Optical wave propagation through the Earth’s atmosphere is governed by spatial and temporal fluctuations in the refractive index. The refractive index structure parameter $C_n^2(z)$ characterizes the local strength of atmospheric turbulence along the propagation path z . The integrated turbulence severity for an optical beam is quantified by the Fried parameter r_0 .¹⁹

2.1.1 Phase screen generation

In numerical wave optics, the continuous turbulent volume is discretized into a sequence of N_s infinitely thin, two-dimensional phase screens placed at discrete propagation distances z_i ,²⁰ see Figure 1 for a graphical illustration. In SPECULA the phase fluctuation $\phi(\mathbf{x})$ added by a screen at position z_i is modeled via its spatial Power Spectral Density (PSD) utilizing the modified von Kármán spectrum.²⁰

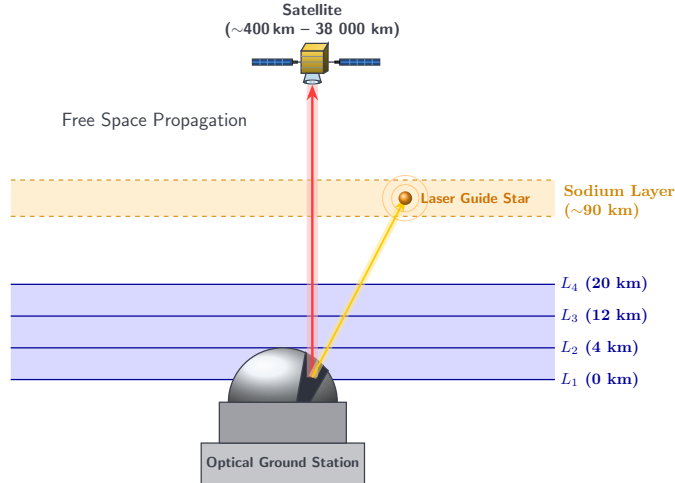


Figure 1: Illustration of atmospheric layers and propagation.

To model the temporal evolution of atmospheric turbulence over the simulation durations, traditional wave optics frameworks often pre-generate and store massive static phase screens in memory, sliding a finite aperture across them at the local wind velocity. However, this approach introduces severe computational bottlenecks, particularly for LEO feeder links. Because LEO satellites traverse the sky rapidly, ground telescope tracking induces high slewing rates that drastically increase the apparent wind speed passing through the beam path. Simulating even a short LEO pass under high effective wind speeds requires pre-generating phase screens tens to hundreds of meters in physical length and demanding prohibitive amounts of memory. To circumvent these limitations, SPECULA implements an infinite phase screen generator.²¹ Rather than allocating a monolithic grid at initialization, the tool dynamically synthesizes new independent phase screen columns on the fly as the turbulence translates across the optical aperture.

2.1.2 Angular Spectrum Method and Fraunhofer Diffraction

For step-by-step propagation between consecutive phase screens separated by a distance $\Delta z = z_{i+1} - z_i$, SPECULA employs the Angular Spectrum Method (ASM).²² The angular spectrum is propagated over distance Δz by applying the transfer function

$$H(\kappa, \Delta z) = \exp \left(jk \Delta z \sqrt{1 - \lambda^2 (\kappa_x^2 + \kappa_y^2)} \right),$$

where $\kappa = (\kappa_x, \kappa_y)$ denotes the spatial frequency vector, λ is the wavelength and k the wave number. The optical field at z_{i+1} prior to the next phase screen is reconstructed via Fourier transforms $\mathcal{F}$ as

$$U(\mathbf{x}, z_{i+1}) = \mathcal{F}^{-1} \{ \mathcal{F} \{ U(\mathbf{x}, z_i) \} \cdot H(\kappa, \Delta z) \}.$$

When evaluating beam propagation across long vacuum distances such as from the top of the atmospheric turbulence layer to a GEO or LEO satellite receiver the propagation distance satisfies the far-field condition $z \gg \frac{k|\mathbf{x}|^2}{2}$, where k denotes the wave number. In this regime, SPECULA transitions from ASM to the Fraunhofer diffraction²² formulation to avoid numerical grid sampling breakdown. The complex field $U(\mathbf{x}, z_{i+1})$ at far-field distance is computed directly as a scaled Fourier transform of the aperture field

$$U(\mathbf{x}, z_{i+1}) = \frac{\exp(jkz_{i+1}) \exp\left(j \frac{k}{2z_{i+1}} |\mathbf{x}|^2\right)}{j\lambda z_{i+1}} \mathcal{F} \{ U(\mathbf{x}, z_i) \}.$$

This dual-propagation strategy, i.e., combining multi-layer ASM through atmospheric turbulence with Fraunhofer transformation for long vacuum propagation, ensures computational accuracy while maintaining FFT sampling constraints across the complete end-to-end optical link.

2.2 Bi-Directional Geometry and Point Ahead Angle

Unlike astronomical AO systems, where the target asset and the reference source often share essentially identical optical paths, bi-directional ground-to-satellite optical links operate under non-reciprocal path geometries. As space assets travel at high orbital velocities relative to the Earth's surface and optical signals propagate at the finite speed of light c , the ground terminal must transmit its uplink beam toward the satellite's future position rather than its apparent position. See Figure 2 for a graphical illustration.

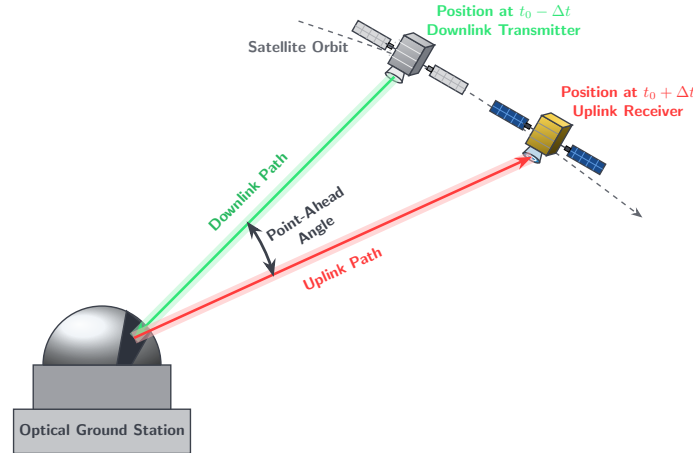


Figure 2: Illustration of bi-directional link geometry.

The apparent position of the satellite observed at the ground station at time t_0 corresponds to the position from which the downlink optical wavefront was emitted at an earlier time $t_0 - \Delta t$. To ensure the uplink laser beam intercepts the satellite payload at time $t_0 + \Delta t$, the ground terminal must point its transmitter ahead along the satellite's velocity vector. The Point-Ahead Angle $\theta_{\text{PAA}}(t)$ is defined as the angular separation between the downlink line-of-sight vector (Figure 2 green) and the required uplink transmission vector (Figure 2 red).⁸ For small angular offsets, θ_{PAA} can be approximated using the transverse component of the relative satellite velocity vector $\mathbf{v}_{\text{rel},\perp}$ by

$$\theta_{\text{PAA}}(t) \approx \frac{2 \mathbf{v}_{\text{rel},\perp}(t)}{c}.$$

For GEO satellites, Earth rotation generates a quasi-static relative velocity giving a nearly constant PAA of about $20 \mu\text{rad}$. For LEO satellites moving at orbital speeds up to 7.8 km/s , $\theta_{\text{PAA}}(t)$ varies dynamically during a pass, reaching up to $50 \mu\text{rad}$ at low elevation angles. As a consequence of the angular displacement, the downlink and uplink wave $U_{\text{up}}(\mathbf{x}, z)$ traverse different atmospheric turbulence volumes. At a propagation distance z along the line of sight, where $z = 0$ denotes the ground aperture and $z = L$ the top of the atmospheric layers, the physical transverse spatial separation $\Delta\mathbf{r}(z, t)$ between the centers of the uplink and downlink beams is expressed as

$$\Delta\mathbf{r}(z, t) = z \cdot \theta_{\text{PAA}}(t).$$

The severity of the resulting point-ahead anisoplanatism depends on how $\Delta\mathbf{r}(z, t)$ compares to the local coherence diameter $r_0(z)$.⁹ When $\Delta\mathbf{r}(z, t) > r_0(z)$, the phase distortions accumulated by the downlink beam become decorrelated from those encountered by the uplink beam. SPECULA models this non-reciprocity by propagating the downlink and uplink optical beams through spatially shifted lateral positions across each numerical phase screen, yielding quantitative predictions of point-ahead degradation on uplink Strehl ratio and power loss.

2.3 Adaptive Optics Loop and Pre-Compensation

AO systems operate by dynamically sensing and correcting phase distortions introduced into an optical wavefront as it propagates through atmospheric turbulence.²³ In a ground-to-satellite optical feeder link, an incoming downlink wavefront is acquired at time step k by the ground aperture and measured by a Wavefront Sensor (WFS). The CCD detector captures an intensity pattern $I(k)$, which a slope computer processes to extract local wavefront slopes $s(k)$. A wavefront reconstructor then converts these slopes into modal commands $m(k)$. A controller, such as a standard leaky integrator, subsequently computes the actuator control signals $c(k)$ applied to the Deformable Mirror (DM), flattening the residual phase profile to maximize Single-Mode Fiber (SMF) coupling efficiency, see Figure 3 for a flow chart and Section 2.3.1 for more details on the reconstructor and control. For uplink transmission, the optical reciprocity principle allows the ground station to leverage these downlink phase measurements to pre-distort the outgoing laser beam. This pre-compensation ensures that the beam flattens as it subsequently traverses the atmosphere, effectively mitigating beam wander and signal fades at the satellite receiver.

A laser can be operated in monostatic or bistatic configuration. Monostatic configurations transmit the uplink laser and receive the downlink signal through a single, shared primary aperture, utilizing components such as a dichroic beamsplitter for path separation. Because both channels share the same optical path, a single DM in the common path can directly pre-distort the uplink beam using measurements from the downlink WFS. In contrast, bistatic configurations decouple the system by launching the uplink beam through a physically separate telescope mounted off-axis or displaced by a certain distance. The lateral separation introduces an additional spatial offset between the pupil apertures, which increases total spatial anisoplanatism when combined with the PAA. SPECULA supports both monostatic and bistatic laser launch configurations, allowing simultaneous modeling of common-path transceiver geometries and separated transmitter-receiver setups.

2.3.1 Wavefront Reconstruction and Control

Managing optical aberrations requires precise wavefront sensing and real-time closed-loop control. Within SPECULA, this process is formalized by spatially discretizing the continuous optical phase. Specifically, the wavefront distortion of the incoming downlink signal is expanded over a finite set of spatial basis functions, such as Zernike or Karhunen–Loève polynomials, yielding the representation $\phi(k) \in \mathbb{R}^{N_\phi}$. The WFS, such as a Shack-Hartmann or Pyramid, samples the phase gradients or intensity distributions across the entrance pupil. The linearized WFS measurement equation yielding the slope vector $\mathbf{s}(k) \in \mathbb{R}^{N_s}$ is modeled as

$$\mathbf{s}(k) = \mathbf{IM} \mathbf{c}(k) + \mathbf{n}(k),$$

where $\mathbf{IM} \in \mathbb{R}^{N_s \times N_c}$ is the interaction matrix linking the N_c deformable mirror actuator commands to the N_s sensor measurements, $\mathbf{c}(k) \in \mathbb{R}^{N_c}$ is the vector of DM actuator commands, and $\mathbf{n}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_n)$ represents WFS measurement noise (combining photon shot noise and detector read noise). The reconstructor

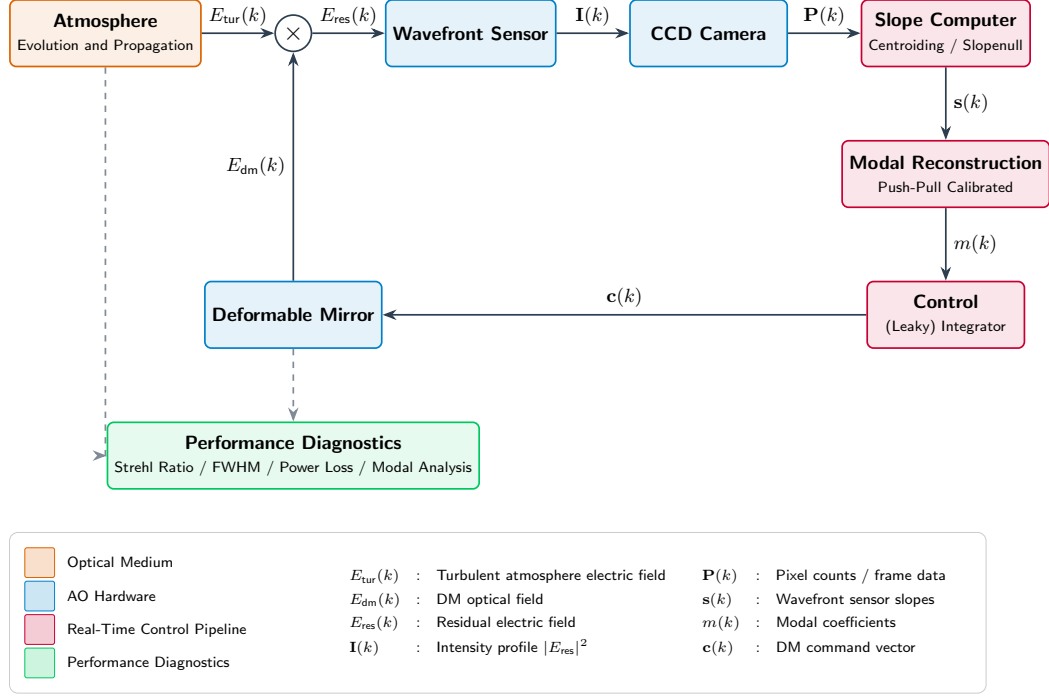


Figure 3: Simulation Overview.

matrix $\mathbf{R} \in \mathbb{R}^{N_c \times N_s}$ estimates the optimal DM actuator commands from noisy slope measurements. SPECULA computes the reconstruction matrix $\mathbf{R}$ by applying Singular Value Decomposition (SVD) to determine the pseudo-inverse of the interaction matrix $\mathbf{IM}$.

In downlink post-compensation, the system operates in a closed loop where the DM continuously counteracts the incoming atmospheric turbulence. The WFS measures the residual phase error, represented by the slope vector $\mathbf{s}(k)$, which results from the difference between the incoming turbulence and the current DM correction. To close the loop, the digital controller uses these residual measurements to compute updated DM commands $\mathbf{c}(k)$ via a discrete-time integral or proportional-integral (PI) control filter. This filter incorporates a loop gain g , a leak factor $\gamma \in (0, 1]$, and an integer frame delay $d \in \mathbb{N}$

$$\mathbf{c}(k) = \gamma \mathbf{c}(k-1) + g \mathbf{R} \mathbf{s}(k-d).$$

Conversely, for uplink pre-compensation, the estimated downlink turbulence is applied to the DM to pre-distort the outgoing laser beam prior to transmission.

2.3.2 Time Delays

The performance of an AO system in a bi-directional optical feeder link is fundamentally constrained by its temporal dynamics. In addition to spatial anisoplanatism, the AO control loop experiences processing pipeline latencies, hardware frame rates, DM dynamics, and finite optical propagation delays. Together, these delays introduce a temporal phase mismatch between the measured downlink wavefront and the executed control command. SPECULA enables the simulation to explicitly account for these time delays. The total latency τ_{total} is governed by several distinct contributions

$$\tau_{\text{total}} = \tau_{\text{wfs}} + \tau_{\text{read}} + \tau_{\text{comp}} + \tau_{\text{dm}} + \tau_{\text{prop}},$$

where τ_{wfs} is the average WFS integration delay, τ_{read} is the detector readout latency, τ_{comp} denotes the RTC processing time, τ_{dm} is the DM mechanical response time and τ_{prop} the one-way optical flight time between the ground station and the space asset.

2.3.3 Link Performance Diagnostics

To quantitatively evaluate the AO performance and optical link budgets across both downlink and uplink paths, SPECULA incorporates performance diagnostics tools. At each time step k , the software processes the propagated complex optical fields and offers to compute core metrics.

Strehl Ratio and Full Width Half Maximum

The Strehl ratio $S(k) \in [0, 1]$ serves as the primary metric for assessing AO correction quality. In SPECULA it is evaluated in the focal plane as the ratio of the central intensity of the PSF of the propagated AO corrected electric field E to the peak intensity of an ideal, diffraction-limited reference beam via

$$S(k) = \frac{|\text{PSF}(\mathbf{0}, k)|}{|\text{PSF}_{\text{diff}}(\mathbf{0})|}, \text{ with } \text{PSF}(\mathbf{0}, k) = |\mathcal{F}\{E(\mathbf{x}, k)\}|_{\mathbf{x}=\mathbf{0}}^2.$$

Because this formulation measures the PSF intensity strictly at the optical axis center $\mathbf{x} = \mathbf{0}$ rather than tracking the instantaneous shifted maximum, it naturally accounts for performance degradation induced by uncorrected tip-tilt.

The Full Width at Half Maximum (FWHM) quantifies the physical spatial extent and spot size of the focal beam. SPECULA computes this metric directly from the PSF of the propagated wavefront, specifically where the curve drops to half of its peak maximum intensity. For an ideal circular pupil, the theoretical diffraction-limited FWHM is given by $1.028 \lambda/D$.²⁴ By comparing this theoretical baseline against the dynamic FWHM, systems can effectively track uncompensated atmospheric effects.

Power Loss of the Uplink Beam

For upwards propagated laser beams, pointing errors due to residual tip-tilt and high-order wavefront aberrations directly degrade the optical power received at the target aperture. SPECULA models this communication power loss $L_{\text{comm}}(k)$ in dB by extracting the central value of the PSF of the upwards propagated electric field $\text{PSF}_{\text{upward}}(\mathbf{0}, k)$, relative to the central peak of the diffraction-limited $\text{PSF}_{\text{diff}}(\mathbf{0})$ via

$$L_{\text{comm}}(k) = 10 \log_{10} \left(\frac{\text{PSF}_{\text{upward}}(\mathbf{0}, k)}{\text{PSF}_{\text{diff}}(\mathbf{0})} \right) \text{ dB}.$$

Modal Analysis

To assess how well the AO loop targets specific spatial frequencies, SPECULA enables to perform a modal analysis. Both the uncorrected phase $\phi_{\text{tur}}(\mathbf{x}, k)$ and the residual phase $\phi_{\text{res}}(\mathbf{x}, k)$ are decomposed onto an orthogonal modal basis, such as Zernike polynomials $Z_i(\mathbf{x})$,

$$\phi(\mathbf{x}, k) = \sum_{i=1}^{N_{\text{modes}}} a_i(k) Z_i(\mathbf{x}).$$

By comparing the uncorrected variance to the residual variance the software measures the correction efficiency for individual aberrations, ranging from tip-tilt and defocus to high-order modes. Note that in wave optics propagation, extracting the phase $\phi(\mathbf{x}, k)$ via $\arg(E(\mathbf{x}, k))$ typically produces a 2π -wrapped phase profile. To obtain a continuous, unwrapped phase surface, SPECULA solves a discrete Poisson equation.²⁵ Compared to traditional path-following unwrapping algorithms, this gradient-based approach provides better numerical stability against branch-cut noise and better manages scintillation-induced phase singularities.

3. SPECULA SOFTWARE ARCHITECTURE

To model complex, multi-scale bi-directional optical links efficiently, the software architecture must balance numerical fidelity and computational complexity. SPECULA^{16,26} is engineered as an open-source*, object-oriented Python package designed specifically for multi-threaded, GPU-accelerated wave optics modeling. To reflect real-world physical ground station architectures, SPECULA decouples system calibration from dynamic time-stepped simulation. The main architectural components are detailed below.

*<https://github.com/ArcetriAdaptiveOptics/SPECULA>

3.1 Modular design and Configuration

SPECULA implements a modular simulation framework structured as a network of interconnected processing objects, which represent physical entities, devices, or processes, such as atmospheric propagation, wavefront sensors, and deformable mirrors. Processing objects communicate exclusively via uniform data objects like slope vectors, actuator commands, and electrical fields. This decoupled architecture allows individual components to be flexibly connected, reconfigured, or swapped. At each simulation time step, a scheduler executes the processing objects in an order that respects the partial-order relations defined by the network topology. Data objects track the temporal metadata of their last modification to maintain synchronization. Each object executes a standardized life-cycle including input collection, computation, and output updating. To eliminate hard-coded parameters and ensure strict reproducibility, the framework is driven entirely by YAML configuration files.

Simulation outputs are typically saved to disk, a process handled explicitly in SPECULA by adding a dedicated processing object to the YAML simulation configuration. Inputs routed to the `DataStore` class are recorded as time series, consisting of the data alongside a time axis, and stored in either FITS format or as serialized Python pickle objects. To monitor simulations, SPECULA provides a suite of visualization processing objects built on the matplotlib library. These are specialized into classes for scalar and vector time series, two-dimensional images, and specialized data objects such as pixels, slopes, and electric fields. Furthermore, SPECULA can host a built-in Flask web server to stream these visualizations over the network.

3.2 Calibration and Interaction Matrix Generation

For physical ground stations, AO control relies on empirical system calibration prior to operation. SPECULA mirrors this real-world workflow by separating execution into an offline calibration phase followed by an online closed-loop simulation. Specifically, the calibration is divided into two main parts.

3.2.1 WFS Calibration

The WFS calibration process is split into the following parts:

- **Valid subaperture determination:** During calibration, SPECULA computes the flux integrated across each subaperture. A subaperture is valid if its total flux exceeds an adjustable intensity threshold. Only valid subapertures are assembled into the active measurement vector, reducing memory overhead and eliminating unilluminated noise from control calculations.
- **Reference Centroid Calibration (slopenull):** Static optical non-common path aberrations or mechanical misalignment introduce constant offsets in the measurement vector. SPECULA records this static slope vector $\mathbf{s}_{\text{null}}$ produced by a flat reference beam passing through the static optics. During dynamic loop execution, this slopenull vector is subtracted from all raw WFS slope measurements $\mathbf{s}_{\text{raw}}(k)$ prior to reconstruction

$$\mathbf{s}(k) = \mathbf{s}_{\text{raw}}(k) - \mathbf{s}_{\text{null}}.$$

This ensures the control loop drives the DM to flatten the wavefront relative to the true optical system zero rather than an uncalibrated hardware offset.

3.2.2 Push-Pull Interaction Matrix Calibration

The push-pull interaction matrix generation method characterizes the linear response of an AO system by sequentially applying positive and negative offsets (a "push" and a "pull") of a known magnitude to each DM actuator or control mode. By measuring the WFS response the method effectively isolates the linear system behavior and constructs an accurate interaction matrix mapping commands to sensor measurements.

To compute the Interaction Matrix $\mathbf{IM} \in \mathbb{R}^{N_s \times N_c}$ linking N_c active deformable mirror actuator degrees of freedom to N_s valid WFS slope measurements, SPECULA applies a differential push-pull sequence to each actuator $j \in \{1, \dots, N_c\}$

$$\mathbf{IM}_j = \frac{\mathbf{s}_j^+ - \mathbf{s}_j^-}{2 a_{\text{push}}},$$

where $\mathbf{s}_j^+$ and $\mathbf{s}_j^-$ represent the slopenull corrected WFS slope vectors measured after applying a positive ($+c_{\text{push}}$) and negative ($-c_{\text{push}}$) poke to actuator j . Once $\mathbf{IM}$ is generated, SPECULA derives the regularized reconstructor matrix $\mathbf{R}$.

3.3 High Performance Computing Support

Combining AO and wave optics simulations including, e.g., multi-layer Fourier propagation and infinite phase screens, results in significant computational loads. SPECULA implements a tiered High-Performance Computing (HPC) acceleration strategy:

- **GPU acceleration via CuPy & CUDA:** Compute-heavy tasks like array operations, fast Fourier transforms, and matrix multiplications are offloaded to GPUs using CuPy and customized CUDA C++ kernels. For maximum throughput, code can be structured into CUDA graphs and executed within dedicated CUDA streams, provided that runtime memory allocations and host-device synchronizations are avoided. Importantly, developers unfamiliar with these low-level CUDA mechanics can still implement standard GPU- and CPU-compatible code without dealing with such complexities.
- **Multi-node distributed scaling via MPI:** SPECULA integrates mpi4py to seamlessly distribute simulations across several computing nodes.

3.4 RTC Integration

When developing software for a real-time control (RTC) system, testing directly on physical hardware is often impractical due to hardware availability, risk, and financial or time constraints. Consequently, simulating as much of the environment as possible is crucial. To bridge SPECULA with RTC systems, we developed dedicated bindings for Artic¹⁷ and DAO.¹⁸ Because Artic is not yet publicly available and the DAO bindings remain a work in progress, neither is currently open source, though a public release is planned soon. These integrations are implemented as specialized SPECULA objects configured directly within the parameter file. They consist of two main class pairs `ArticSource/ArticShm` and their DAO counterparts, `DaoSource/DaoShm`:

- **ArticShm:** This class receives input from other SPECULA components (most commonly the CCD) and writes it to a shared memory region. Users can configure the target output alongside basic operations such as type conversion, image resizing, and scaling.
- **ArticSource:** This class reads from a designated shared memory region and outputs the result for other SPECULA components. Typically, it reads the computed command vector from the RTC and feeds it into the SPECULA deformable mirror (DM) class. Additional processing options, such as scaling and index selection, are available, alongside adapters to interface with various other Artic shared memory structures.

The combined RTC simulation workflow is illustrated in Figure 4. SPECULA simulates atmospheric turbulence and computes the resulting wavefront sensor measurements on the CCD. These measurements are then written to the RTC shared memory via `ArticShm`. Within the RTC, a dedicated component reads frames from the shared memory region and forwards them down the pipeline, mimicking the behavior of a real camera during live operations. The RTC performs its standard computations, while `ArticSource` waits for the RTC to finish and write the resulting command vector back to shared memory. Finally, the simulation applies the new command, computes the DM shape, and generates the subsequent measurements.

Integrating the RTC into a simulation requires minor modifications. Input frames are sourced from shared memory instead of camera hardware, and output commands are stored in shared memory rather than transmitted to a physical DM, though this is standard practice on our RTC for diagnostics anyway. Timeouts on the RTC may have to be more lenient, because simulations often struggle to meet real-time requirements. To test for performance bottlenecks, we implemented a non-waiting mode in the camera component that repeatedly sends the same simulated frame at a fixed operational framerate, instead of waiting for the next simulated frame. Naturally, this approach requires careful handling of side effects, for instance, control loop gains must be adjusted accordingly to account for frame repetition.

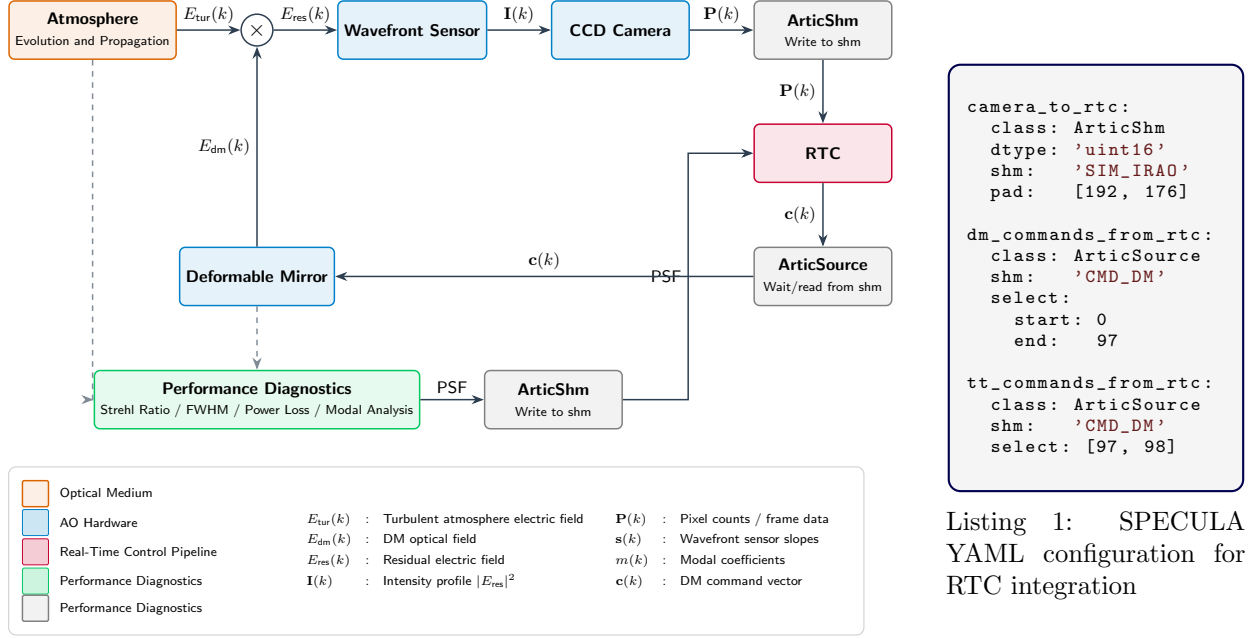


Figure 4: Simulation with RTC bindings.

3.5 Software Validation

Figure 5 illustrates the comprehensive validation workflow for the SPECULA framework, combining an automated internal integration pipeline with external verification methods. Internally, the software is validated through a comprehensive test suite. This includes unit tests to verify individual software functionalities, integration tests to ensure correct interaction between modules, and validation tests that compare component outputs against reference data. These tests run automatically upon code modifications within a continuous integration framework.

To ensure scientific validity, SPECULA has been systematically benchmarked against PASSATA,¹⁰ a well-established simulation library widely trusted in astronomical applications and verified across diverse contexts ranging from complex instrument design to on-sky commissioning.^{27–31} Furthermore, SPECULA has undergone cross-verification with the semi-analytical simulator FAST¹⁵ within the collaborative framework of the ALASCA project,^{32,33} which focuses on the design and implementation of an optical feeder link using LGS-AO at the ESA’s optical ground station in Tenerife. As part of this validation pipeline, early open-loop telemetry data gathered from the ALASCA testbed facility in Tenerife served as an empirical baseline for comparing simulated outputs against real-world measurements, with broader telemetry datasets planned for integration as the project continues to evolve.

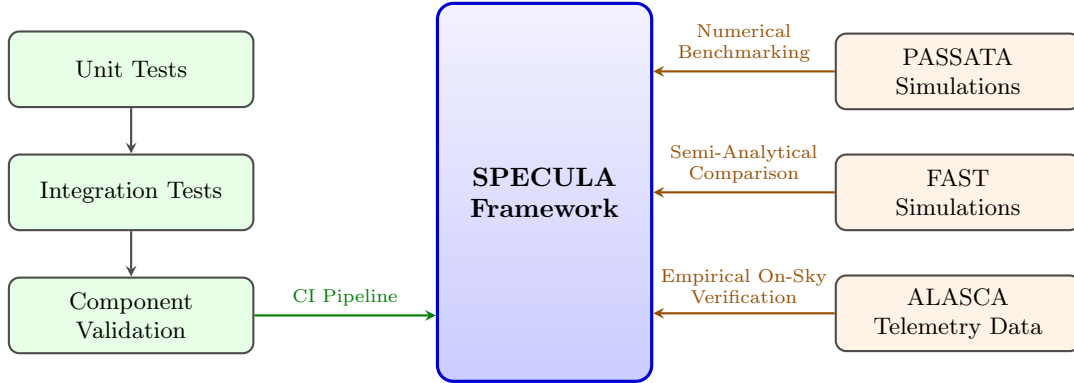


Figure 5: Structure of the SPECULA verification and validation workflow.

4. SIMULATION CAPABILITIES AND NUMERICAL EXAMPLES

To illustrate SPECULA's simulation possibilities, this section presents numerical examples for physical propagation and an optical feeder link scenario. Note that some YAML code excerpts are shown here, the full tutorial can be found in the SPECULA GitHub repository.

4.1 Case Study 1: Optical Wave Propagation Through Turbulence

This case study presents a numerical simulation modeling the propagation of an optical wave through turbulent media. By replicating atmospheric conditions, such as varying turbulence strengths parameterized by the refractive index structure constant C_n^2 , wind and atmospheric seeing, this SPECULA simulation investigates the degradation of optical wavefronts. The framework accounts for time delays in upwards and downwards propagation, enabling the precise analysis of bi-directional link dynamics.

```
main:
  root_dir:      './calib/'
  class:         'SimulParams'
  pixel_pupil:   120
  pixel_pitch:   0.008333
  total_time:    5.0
  time_step:     0.0005
  zenithAngleInDeg: 56.0

pupilstop:
  class:         'Pupilstop'
  simul_params_ref: 'main'
  mask_diam:     1.0
```

Listing 2: SPECULA YAML configuration for general telescope parameters.

```
seeing:
  class:         'WaveGenerator'
  constant:      1.6
  outputs:       ['output']

wind_speed:
  class:         'WaveGenerator'
  constant:      [3.54, 84.94, 387.74]
  outputs:       ['output']

wind_direction:
  class:         'WaveGenerator'
  constant:      [90., 270., 0.]
  outputs:       ['output']

atmo:
  class:         'AtmoInfiniteEvolutionUpDown'
  simul_params_ref: 'main'
  L0:            20.0
  heights:       [0, 4000, 20000]
  Cn2:           [0.95, 0.01, 0.04]
  extra_delta_time_down: [0, 0.00002, 0.0001]
  extra_delta_time_up:  [0.00045, 0.00042, 0.00033]
  seed:          1
  fov_in_m:      8
  inputs:
    seeing:      'seeing.output'
    wind_speed:  'wind_speed.output'
    wind_direction: 'wind_direction.output'
  outputs:       ['layer_list_down',
                  'layer_list_up']
```

Listing 3: SPECULA YAML configuration for atmospheric turbulence.

```
source_up:
  class:         'Source'
  polar_coordinates: [0.0, 0.0]
  magnitude:      0
  wavelengthInNm: 1550
  height:         400000

source_down:
  class:         'Source'
  polar_coordinates: [5.0, 180.0]
  magnitude:      0
  wavelengthInNm: 1064
  height:         400000

prop_down:
  class:         'AtmoPropagation'
  simul_params_ref: 'main'
  source_dict_ref: ['source_down']
  doFresnel:      true
  padding_factor: 4
  wavelengthInNm: 1064
  inputs:
    atmo_layer_list: ['atmo.layer_list']
    common_layer_list: ['pupilstop']
  outputs:        ['out_source_down_ef']

prop_up:
  class:         'AtmoPropagation'
  simul_params_ref: 'main'
  source_dict_ref: ['source_up']
  doFresnel:      true
  upwards:        true
  padding_factor: 4
  wavelengthInNm: 1550
  inputs:
    atmo_layer_list: ['atmo.layer_list']
    common_layer_list: ['pupilstop']
  outputs:        ['out_source_up_ef']
```

Listing 4: SPECULA YAML configuration for bi-directional propagation parameters.

Listing 2 outlines the general simulation parameters, including a zenith angle of 56° , a total simulation time of 5 s, a frame rate of 2 kHz, a resolution of 120×120 pixels with a pixel pitch of 0.00833 m. The `pupilstop` block defines a circular telescope pupil with 1 m diameter used for both, upwards and downwards propagation, though alternative geometries, such as a Gaussian profile for uplink laser beams, can also be configured. Additionally, the `root_dir` specifies the directory for storing calibration-generated files.

Listing 3 shows an excerpt of the SPECULA YAML configuration file dedicated to atmospheric turbulence, where the `AtmoInfiniteEvolutionUpDown` class generates three turbulent layers using the infinite phase screen method (refer to Section 2.1.1 for details). Wind speeds are scaled appropriately to account for telescope slewing, while the `extra_delta_time_up` and `extra_delta_time_down` parameters account for temporal propagation delays across the bi-directional paths.

Listing 4 illustrates the YAML configuration for wave propagation to and from a space object at an altitude of 400 km, utilizing uplink and downlink wavelengths of 1550 nm and 1064 nm, respectively. To account for PAA effects, the downlink source is positioned 5.0 arcsec off-axis. Physical optics propagation via ASM or Fraunhofer approximation is enabled by setting `doFresnel=true` and matching the wavelength to the defined source. To mitigate circular wrapping artifacts arising from the Fast Fourier Transforms (FFTs) inherent to ASM and Fraunhofer implementations, a padding factor of 4 times the input size is applied. Finally, while phase extraction following these propagation methods naturally yields a 2π -wrapped field, SPECULA incorporates a dedicated 2D phase unwrapping tool based on the Poisson equation to reconstruct continuous, smooth phase maps.

4.2 Case Study 2: Optical Feeder Link

Building upon the atmospheric and propagation setups established in the previous section, this case study presents an optical feeder link scenario for a LEO satellite at an altitude of 400 km, featuring a PAA of 5 arcsec between the uplink and downlink paths. A monostatic system is used, i.e., the uplink laser is transmitted through the same aperture as the downlink is received. The downlink signal, in conjunction with a Pyramid WFS, estimates the turbulent wavefront, which is subsequently used to drive a DM via an integrator controller. The interaction and reconstruction matrices are established through a calibration procedure utilizing a push-pull approach. The DM corrects the downlink path in closed-loop while simultaneously pre-compensating the uplink laser beam.

Listing 5 details the WFS setup, where the downlink electric field is fed into the Pyramid WFS and a simulated CCD detector measures the resulting intensity pattern to compute the local slopes.

Listing 6 presents the reconstruction, control, and DM configurations. The calculated slope vector is multiplied by the reconstruction matrix `rec_mat`, derived from the push-pull calibration, and processed by an integrator controller using tailored modal gains to generate the commands required to actuate the DM. Unlike open-loop setups, running closed-loop simulations requires explicitly integrating the DM into the propagation path using the `common_layer_list` parameter for both `source_up` and `source_down`, enabling downlink post-compensation and uplink pre-compensation. The overall performance is evaluated via the `power_loss` of the uplink laser beam.

Figure 6 shows a histogram of the uplink power loss derived from the SPECULA configuration files, illustrating the performance gains achieved by employing AO, which reduces the mean power loss from -16.7 dB to -3.3 dB. This 5 s simulation takes about 6 minutes to run on an NVIDIA A100 GPU utilizing SPECULA’s GPU acceleration via CuPY and CUDA. Note that for the non AO simulation, the `int_gains` parameter in the `control` block is set to 0. To capture these results, a dedicated `data_store` block was integrated into the parameter file to log the output of `power_loss`.

More detailed instructions are provided in the tutorials available in the SPECULA documentation[†].

[†]<https://specula.readthedocs.io/>

```

pyramid:
  class: 'ModulatedPyramid'
  simul_params_ref: 'main'
  pup_diam: 40.
  pup_dist: 72.
  fov: 5.0
  fov_errinf: 0.1
  fov_errsup: 30.0
  mod_amp: 2.4
  output_resolution: 240
  fft_res: 3.0
  wavelengthInNm: 1064
  inputs:
    in_ef: 'prop_down.out_source_down_ef'
  outputs:
    out_i: ['out_i']

detector:
  class: 'CCD'
  simul_params_ref: 'main'
  size: [240,240]
  dt: 0.001
  bandw: 10
  photon_noise: true
  excess_noise: true
  readout_noise: true
  readout_level: 0.5
  inputs:
    in_i: 'pyramid.out_i'
  outputs:
    out_pixels: ['out_pixels']

slopec:
  class: 'PyrSlopec'
  pupdata_object: 'pupdata'
  thr_value: 0.0
  inputs:
    in_pixels: 'detector.out_pixels'
  outputs:
    out_slopes: ['out_slopes']

```

Listing 5: SPECULA YAML configuration for Pyramid WFS.

```

modalrec:
  class: 'Modalrec'
  recmat_object: 'rec_mat'
  inputs:
    in_slopes: 'slopec.out_slopes'
  outputs:
    out_modes: ['out_modes']

control:
  class: 'Integrator'
  delay: 1
  int_gain: [0.5,0.4]
  n_modes: [2,57]
  inputs:
    delta_comm: 'modalrec.out_modes'
  outputs:
    out_comm: ['out_comm']

dm:
  class: 'DM'
  simul_params_ref: 'main'
  type: 'zernike'
  height: 0
  nmodes: 59
  inputs:
    in_command: 'control.out_comm'
  outputs:
    out_layer: ['out_layer']

power_loss:
  class: 'PowerLoss'
  simul_params_ref: 'main'
  prop_ref: 'prop_up'
  inputs:
    in_ef: 'prop_up.out_source_up_ef'
  outputs:
    out_power_loss: ['out_power_loss']

```

Listing 6: SPECULA YAML configuration for modal reconstruction, control and DM.

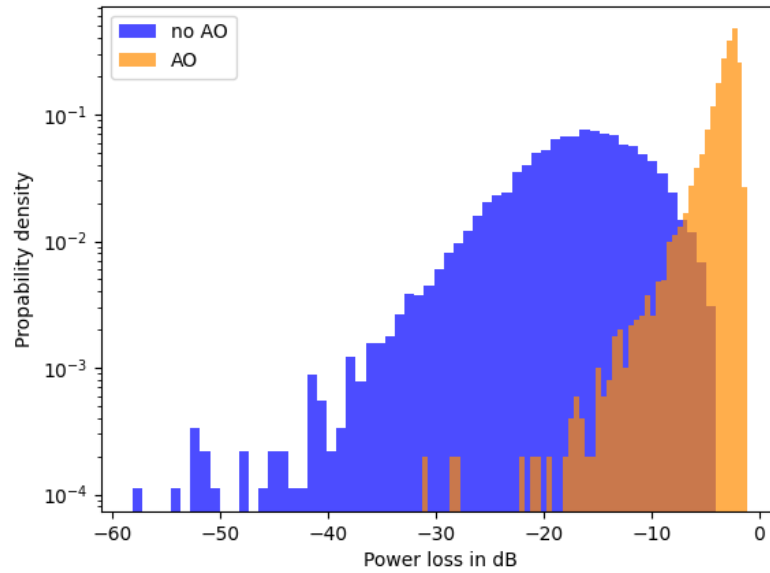


Figure 6: Histogram of the power loss of the uplink laser beam.

5. CONCLUSION

In this work, we have demonstrated the efficacy of the SPECULA framework for modeling bi-directional optical feeder links, bridging the gap between high numerical fidelity and computational performance. By implementing a modular, object-oriented architecture that separates static system calibrations from dynamic, time-stepped simulations, SPECULA provides a flexible environment for analyzing wave optics scenarios under realistic ground-station constraints. As space-to-ground optical communication networks transition toward higher data rates and increasingly demanding architectures, scalable and GPU-accelerated simulation tools like SPECULA will be crucial for optimizing future optical feeder link designs and validating adaptive optics performance prior to deployment.

ACKNOWLEDGMENTS

The authors were supported by the NVIDIA Corporation Academic Hardware Grant Program. B. Stadler is partially supported by the State of Upper Austria. R. Ramlau is funded in part by the Austrian Science Fund (FWF) SFB 10.55776/F68 “Tomography Across the Scales”, project F6805-N36 (Tomography in Astronomy).

REFERENCES

- [1] Toyoshima, M., “Trends in satellite communications and the role of optical free-space communications,” *Journal of Optical Networking* (2005).
- [2] Williams, W. D., Collins, M., Boroson, D. M., Lesh, J., Biswas, A., Orr, R., Schuchman, L., and Sands, O. S., “RF and Optical Communications: A Comparison of High Data Rate Returns From Deep Space in the 2020 Timeframe,” (2007).
- [3] Boddeda, R., Arrieta, D. R., Almonacil, S., Renaudier, J., and Bigo, S., “Achievable Capacity of Geostationary-Ground Optical Links,” *Journal of Lightwave Technology* **41**, 3717–3725 (6 2023).
- [4] Kaushal, H. and Kaddoum, G., “Optical Communication in Space: Challenges and Mitigation Techniques,” (1 2017).
- [5] Liao, S. K., Cai, W. Q., Liu, W. Y., Zhang, L., Li, Y., Ren, J. G., Yin, J., Shen, Q., Cao, Y., Li, Z. P., Li, F. Z., Chen, X. W., Sun, L. H., Jia, J. J., Wu, J. C., Jiang, X. J., Wang, J. F., Huang, Y. M., Wang, Q., Zhou, Y. L., Deng, L., Xi, T., Ma, L., Hu, T., Zhang, Q., Chen, Y. A., Liu, N. L., Wang, X. B., Zhu, Z. C., Lu, C. Y., Shu, R., Peng, C. Z., Wang, J. Y., and Pan, J. W., “Satellite-to-ground quantum key distribution,” *Nature* **549**, 43–47 (9 2017).
- [6] Shapiro, J. H., “Reciprocity of the Turbulent Atmosphere*,” *J. Opt. Soc. Am.* **61**, 492–495 (4 1971).
- [7] Lutomirski, R. F. and Yura, H. T., “Propagation of a Finite Optical Beam in an Inhomogeneous Medium,” *Appl. Opt.* **10**, 1652–1658 (7 1971).
- [8] Andrews, L. C. and Phillips, R. L., [*Laser Beam Propagation through Random Media*], vol. PM152 of *SPIE Press Monograph*, SPIE Press, Bellingham, WA, USA, 2nd ed. (2005).
- [9] Fried, D. L. and Tyler, G. A., “Strategic laser communications uplink analysis,” technical report, Defense Technical Information Center, Fort Belvoir, VA, USA (1981).
- [10] Agapito, G., Puglisi, A., and Esposito, S., “Passata - object oriented numerical simulation software for adaptive optics,” in [*Adaptive Optics Systems V*], **9909**, 99097E, International Society for Optics and Photonics (2016).
- [11] Conan, R. and Correia, C., “Object-oriented matlab adaptive optics toolbox,” in [*Adaptive Optics Systems IV*], **9148**, 91486C, International Society for Optics and Photonics (2014).
- [12] Ferreira, F., Gratadour, D., Sevin, A., and Doucet, N., “Compass: An efficient gpu-based simulation software for adaptive optics systems,” in [*2018 International Conference on High Performance Computing Simulation (HPCS)*], 180–187, IEEE (2018).
- [13] H  ritier, C. T., Neichel, B., Fusco, T., Wildi, F., and Sauvage, J.-F., “Object-oriented python adaptive optics (oopao): A flexible framework for end-to-end adaptive optics simulations,” in [*Adaptive Optics Systems IX*], **13097**, 1309710, International Society for Optics and Photonics (2024).
- [14] Reeves, A., “Soapy: an adaptive optics simulation written purely in python for rapid concept development,” in [*Adaptive Optics Systems V*], **9909**, 99097F, International Society for Optics and Photonics (2016).

- [15] Farley, O. J. D., Townson, M. J., and Osborn, J., “Fast: Fourier domain adaptive optics simulation tool for bidirectional ground-space optical links through atmospheric turbulence,” *Optics Express* **30**(13), 23050–23064 (2022).
- [16] Rossi, F., Puglisi, A., and Agapito, G., “Introducing a new generation adaptive optics simulation framework: from PASSATA to SPECULA,” *Journal of Astronomical Telescopes, Instruments, and Systems* **12**(1), 019001 (2026).
- [17] Jenkins, D., “Advanced real-time instrument control (artic).”
- [18] Barr, D., Cetre, S., Connolly, J., and Davies, T., “Durham-adaptive-optics/daobase: Initial release,” (Oct. 2025).
- [19] Fried, D. L., “Statistics of a geometric representation of wavefront distortion,” *Journal of the Optical Society of America* **55**(11), 1427–1435 (1965).
- [20] Roggemann, M. C., Welsh, B. M., Montera, D., and Rhoadarmer, T. A., “Method for simulating atmospheric turbulence phase effects for multiple time slices and anisoplanatic conditions,” *Applied Optics* **34**(20), 4037–4051 (1995).
- [21] Assémat, F., Wilson, R., and Gendron, E., “Method for simulating infinitely long and non stationary phase screens with optimized memory storage,” *Optics Express* **14**(3), 988–999 (2006).
- [22] Goodman, J. W., [*Introduction to Fourier Optics*], Roberts and Company Publishers, Englewood, Colorado, 3rd ed. (2005).
- [23] Tyson, R. K., [*Principles of Adaptive Optics*], CRC Press, 4th ed. (2015).
- [24] Born, M. and Wolf, E., [*Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*], Cambridge University Press, 7th (expanded) ed. (1999).
- [25] Zhao, Z., Zhang, H., Ma, C., Fan, C., and Zhao, H., “Comparative study of phase unwrapping algorithms based on solving the poisson equation,” *Measurement Science and Technology* **31**(6), 065004 (2020).
- [26] Rossi, F., Agapito, G., Puglisi, A. T., Stadler, B., and et al, “SPECULA: adoption, developments, and use cases of a scalable adaptive optics simulation framework,” in [*Proc. SPIE 14150, Adaptive Optics Systems X*], **14150** (2026).
- [27] Agapito, G. and Pinna, A., “Semianalytical error budget for adaptive optics systems with pyramid wavefront sensors,” *Journal of Astronomical Telescopes, Instruments, and Systems* **5**(4), 049001 (2019).
- [28] Neichel, B., Agapito, G., Kuznetsov, A., et al., “TipTop: toward a single tool for all ELT instrument’s PSF prediction,” in [*Adaptive Optics Systems IX*], Jackson, K. J., Schmidt, D., and Vernet, E., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **13097**, 130972Y, SPIE (2024).
- [29] Pinna, E., Rossi, F., Puglisi, A., et al., “Bringing SOUL on sky,” in [*Adaptive Optics for Extremely Large Telescopes (AO4ELT6)*], (2019).
- [30] Riccardi, A., Puglisi, A., Grani, P., et al., “The ERIS Adaptive Optics System: first on-sky results of the ongoing commissioning at the VLT-UT4,” in [*Adaptive Optics Systems VIII*], Schreiber, L., Schmidt, D., and Vernet, E., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **12185**, 1218508, SPIE (2022).
- [31] Esposito, S., Riccardi, A., Pinna, E., et al., “Large Binocular Telescope Adaptive Optics System: new achievements and perspectives in adaptive optics,” in [*Astronomical Adaptive Optics Systems and Applications IV*], Tyson, R. K. and Hart, M., eds., *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series* **8149**, 814902, SPIE (2011).
- [32] Biasi, R., Calia, D. B., Centrone, M., Enderlein, M., Faccini, M., Farley, O. J., Gallieni, D., Gooding, D., Haguenauer, P., Janout, P., Jenkins, D., Lison, F., Luis, J., Rey, N. M., Osborn, J., Patauner, C., Pescoller, D., Reyes, M., Sabatini, L., Salvi, L., Sodnik, Z., Alaluf, D., Tintori, M., Tordi, M., Townson, M., Virdee, H., Wei, D., and Clements, W. R., “ALASCA: the ESA Laser Guide Star Adaptive Optics Optical Feeder Link demonstrator facility,” in [*International Conference on Space Optics — ICSO 2022*], Sodnik, Z., Cugny, B., and Haase, N., eds., **12777**, 1277756, SPIE (2023).
- [33] Rey, N. M., Agapito, G., Stadler, B., Calia, D. B., Farley, O. J. D., Jenkins, D., Alaluf, D., Reyes, M., Centrone, M., and Puglisi, A., “Using PASSATA for the numerical simulations of the CaNaPy LGS-AO monostatic, pre-compensated system,” in [*Adaptive Optics for Extremely Large Telescopes (AO4ELT7)*], (2023).